Package 'cicalc'

July 21, 2025

Type Package

Title Calculate Confidence Intervals

Version 0.1.0

Description This calculates a variety of different CIs for proportions and difference of proportions that are commonly used in the pharmaceutical industry including Wald, Wilson, Clopper-Pearson, Agresti-Coull and Jeffreys for proprotions. And Miettinen-Nurminen (1985) <doi:10.1002/sim.4780040211>, Wald, Haldane, and Mee <https://www.lexjansen.com/wuss/2016/127_Final_Paper_PDF.pdf> for difference in proportions.

License Apache License (>= 2)

URL https://gsk-biostatistics.github.io/cicalc/

Depends R (>= 4.1.0)

Imports broom, cli, dplyr, forcats, glue, purrr, rlang, tidyr

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Encoding UTF-8

RoxygenNote 7.3.2

NeedsCompilation no

Author Christina Fillmore [aut, cre] (ORCID: <https://orcid.org/0000-0003-0595-2302>), GlaxoSmithKline Research & Development Limited [cph, fnd], Mike Sprys [aut], Dan Lythgoe [aut] (ORCID: <https://orcid.org/0000-0002-4682-772X>)

Maintainer Christina Fillmore <christina.e.fillmore@gsk.com>

Repository CRAN

Date/Publication 2025-07-21 08:50:06 UTC

Contents

`ci_prop_agresti_coull` *Agresti-Coull CI*

Description

Calculates the `Agresti-Coull` interval (created by `Alan Agresti` and `Brent Coull`) by (for 95% CI) adding two successes and two failures to the data and then using the Wald formula to construct a CI.

Usage

```
ci_prop_agresti_coull(x, conf.level = 0.95, data = NULL)
```

Arguments

`x`	(`binary/numeric/logical`) vector of a binary values, i.e. a logical vector, or numeric with values `c(0, 1)`
`conf.level`	(`scalar numeric`) a scalar in (0,1) indicating the confidence level. Default is 0.95
`data`	(`data.frame`) Optional data frame containing the variables specified in `x` and `by`.

Details

$$\left(\frac{\tilde{p} + z^2_{\alpha/2}/2}{n + z^2_{\alpha/2}} \pm z_{\alpha/2} \sqrt{\frac{\tilde{p}(1-\tilde{p})}{n} + \frac{z^2_{\alpha/2}}{4n^2}} \right)$$

Value

An object containing the following components:

`n`	Number of responses
`N`	Total number
`estimate`	The point estimate of the proportion
`conf.low`	Lower bound of the confidence interval
`conf.high`	Upper bound of the confidence interval
`conf.level`	The confidence level used
`method`	Type of method used

`ci_prop_clopper_pearson`

Clopper-Pearson CI

Description

Calculates the Clopper-Pearson interval by calling `stats::binom.test()`. Also referred to as the `exact` method.

Usage

```
ci_prop_clopper_pearson(x, conf.level = 0.95, data = NULL)
```

Arguments

`x`	(`binary/numeric/logical`) vector of a binary values, i.e. a logical vector, or numeric with values `c(0, 1)`
`conf.level`	(`scalar numeric`) a scalar in (0,1) indicating the confidence level. Default is 0.95
`data`	(`data.frame`) Optional data frame containing the variables specified in `x` and `by`.

Details

$$\left(\frac{k}{n} \pm z_{\alpha/2}\sqrt{\frac{\frac{k}{n}(1-\frac{k}{n})}{n} + \frac{z^2_{\alpha/2}}{4n^2}}\right) / \left(1 + \frac{z^2_{\alpha/2}}{n}\right)$$

Value

An object containing the following components:

n	Number of responses
N	Total number
estimate	The point estimate of the proportion
conf.low	Lower bound of the confidence interval
conf.high	Upper bound of the confidence interval
conf.level	The confidence level used
method	Type of method used

`ci_prop_diff_haldane` *Haldane Confidence Interval for Difference in Proportions*

Description

Haldane Confidence Interval for Difference in Proportions

Usage

```
ci_prop_diff_haldane(x, by, conf.level = 0.95, data = NULL)
```

Arguments

x	(binary/numeric/logical) vector of a binary values, i.e. a logical vector, or numeric with values `c(0, 1)`
by	(string) A character or factor vector with exactly two unique levels identifying the two groups to compare. Can also be a column name if a data frame provided in the `data` argument.
conf.level	(scalar numeric) a scalar in (0,1) indicating the confidence level. Default is 0.95
data	(data.frame) Optional data frame containing the variables specified in `x` and `by`.

Details

The confidence interval is calculated by $\theta^* \pm w$ where:

$$\theta^* = \frac{(\hat{p}_1 - \hat{p}_2) + z^2 v(1 - 2\hat{\psi})}{1 + z^2 u}$$

where

$$w = \frac{z}{1 + z^2 u}\sqrt{u\{4\hat{\psi}(1-\hat{\psi}) - (\hat{p}_1 - \hat{p}_2)^2\} + 2v(1 - 2\hat{\psi})(\hat{p}_1 - \hat{p}_2) + 4z^2 v^2 (1 - 2\hat{\psi})^2}$$

$$\hat{\psi} = \frac{\hat{p}_1 + \hat{p}_2}{2}$$

$$u = \frac{1/n_1 + 1/n_2}{4}$$

$$v = \frac{1/n_1 - 1/n_2}{4}$$

Value

An object containing the following components:

`n`	The number of responses for each group
`N`	The total number in each group
`estimate`	The point estimate of the difference in proportions (theta*)
`conf.low`	Lower bound of the confidence interval
`conf.high`	Upper bound of the confidence interval
`conf.level`	The confidence level used
`method`	Haldane Confidence Interval

References

Constructing Confidence Intervals for the Differences of Binomial Proportions in SAS

Examples

```
responses <- expand(c(9, 3), c(10, 10))
arm <- rep(c("treat", "control"), times = c(10, 10))

# Calculate 95% confidence interval for difference in proportions
ci_prop_diff_haldane(x = responses, by = arm)
```

`ci_prop_diff_jp` *Jeffreys-Perks Confidence Interval for Difference in Proportions*

Description

Jeffreys-Perks Confidence Interval for Difference in Proportions

Usage

```
ci_prop_diff_jp(x, by, conf.level = 0.95, data = NULL)
```

Arguments

x	(binary/numeric/logical) vector of a binary values, i.e. a logical vector, or numeric with values c(0, 1)
by	(string) A character or factor vector with exactly two unique levels identifying the two groups to compare. Can also be a column name if a data frame provided in the data argument.
conf.level	(scalar numeric) a scalar in (0,1) indicating the confidence level. Default is 0.95
data	(data.frame) Optional data frame containing the variables specified in x and by.

Details

The confidence interval is calculated by $\theta^* \pm w$ where:

$$\theta^* = \frac{(\hat{p}_1 - \hat{p}_2) + z^2 v(1 - 2\hat{\psi})}{1 + z^2 u}$$

where

$$w = \frac{z}{1 + z^2 u}\sqrt{u\{4\hat{\psi}(1-\hat{\psi}) - (\hat{p}_1 - \hat{p}_2)^2\} + 2v(1 - 2\hat{\psi})(\hat{p}_1 - \hat{p}_2) + 4z^2 v^2 (1 - 2\hat{\psi})^2}$$

$$\hat{\psi} = \frac{1}{2}\left(\frac{x_1 + 1/2}{n_1 + 1} + \frac{x_2 + 1/2}{n_2 + 1}\right)$$

$$u = \frac{1/n_1 + 1/n_2}{4}$$

$$v = \frac{1/n_1 - 1/n_2}{4}$$

Value

An object containing the following components:

n	The number of responses for each group
N	The total number in each group
estimate	The point estimate of the difference in proportions (theta*)
conf.low	Lower bound of the confidence interval
conf.high	Upper bound of the confidence interval
conf.level	The confidence level used
method	Jeffreys-Perks Confidence Interval

References

Constructing Confidence Intervals for the Differences of Binomial Proportions in SAS

Examples

```
responses <- expand(c(9, 3), c(10, 10))
arm <- rep(c("treat", "control"), times = c(10, 10))

# Calculate 95% confidence interval for difference in proportions
ci_prop_diff_jp(x = responses, by = arm)
```

`ci_prop_diff_mee` *Mee Confidence Interval for Difference in Proportions*

Description

Mee Confidence Interval for Difference in Proportions

Usage

```
ci_prop_diff_mee(x, by, conf.level = 0.95, delta = NULL, data = NULL)
```

Arguments

`x`	(`binary/numeric/logical`) vector of a binary values, i.e. a logical vector, or numeric with values `c(0, 1)`
`by`	(`string`) A character or factor vector with exactly two unique levels identifying the two groups to compare. Can also be a column name if a data frame provided in the `data` argument.
`conf.level`	(`scalar numeric`) a scalar in (0,1) indicating the confidence level. Default is 0.95
`delta`	(`numeric`) Optionally a single number or a vector of numbers between -1 and 1 (not inclusive) to set the difference between two groups under the null hypothesis. If provided, the function returns the test statistic and p-value under the `delta` hypothesis.
`data`	(`data.frame`) Optional data frame containing the variables specified in `x` and `by`.

Details

The confidence interval is calculated by $\theta^* \pm w$ where:

$$\theta^* = \frac{(\hat{p}_1 - \hat{p}_2) + z^2 v(1 - 2\hat{\psi})}{1 + z^2 u}$$

where

$$w = \frac{z}{1 + z^2 u}\sqrt{u\{4\hat{\psi}(1-\hat{\psi}) - (\hat{p}_1 - \hat{p}_2)^2\} + 2v(1 - 2\hat{\psi})(\hat{p}_1 - \hat{p}_2) + 4z^2 v^2 (1 - 2\hat{\psi})^2}$$

$$\hat{\psi} = \frac{1}{2}\left(\frac{x_1 + 1/2}{n_1 + 1} + \frac{x_2 + 1/2}{n_2 + 1}\right)$$

$$u = \frac{1/n_1 + 1/n_2}{4}$$

$$v = \frac{1/n_1 - 1/n_2}{4}$$

Value

An object containing the following components:

`n`	The number of responses for each group
`N`	The total number in each group
`estimate`	The point estimate of the difference in proportions (p1-p2)
`conf.low`	Lower bound of the confidence interval
`conf.high`	Upper bound of the confidence interval
`conf.level`	The confidence level used
`method`	Mee Confidence Interval

References

Constructing Confidence Intervals for the Differences of Binomial Proportions in SAS

Examples

```
responses <- expand(c(9, 3), c(10, 10))
arm <- rep(c("treat", "control"), times = c(10, 10))

# Calculate 95% confidence interval for difference in proportions
ci_prop_diff_mee(x = responses, by = arm)
```

`ci_prop_diff_mn` *Miettinen-Nurminen Confidence Interval for Difference in Proportions*

Description

Calculates the Miettinen-Nurminen (MN) confidence interval for the difference between two proportions. This method can be more accurate than traditional methods, especially with small sample sizes or proportions close to 0 or 1.

Usage

```
ci_prop_diff_mn(x, by, conf.level = 0.95, delta = NULL, data = NULL)
```

Arguments

x	(binary/numeric/logical) vector of a binary values, i.e. a logical vector, or numeric with values c(0, 1)
by	(string) A character or factor vector with exactly two unique levels identifying the two groups to compare. Can also be a column name if a data frame provided in the data argument.
conf.level	(scalar numeric) a scalar in (0,1) indicating the confidence level. Default is 0.95
delta	(numeric) Optionally a single number or a vector of numbers between -1 and 1 (not inclusive) to set the difference between two groups under the null hypothesis. If provided, the function returns the test statistic and p-value under the delta hypothesis.
data	(data.frame) Optional data frame containing the variables specified in x and by.

Details

The function implements the Miettinen-Nurminen method to compute confidence intervals for the difference between two proportions. This approach:

- Calculates the Miettinen-Nurminen score test statistic for different possible values of the proportion difference (delta)
- Identifies the delta values where the test statistic equals the critical value corresponding to the desired confidence level
- Returns these boundary values as the confidence interval limits

The method uses a score test with a small-sample correction factor, making it more accurate than normal approximation methods, especially for small samples or extreme proportions. The equation for the test statistics is as follows:

$$H_0 : \hat{d} - \delta <= 0 \qquad \text{vs.} \qquad H_1 : \hat{d} - \delta > 0$$

$$T_\delta = \frac{\hat{p_x} - \hat{p_y} - \delta}{\sigma_{mn}(\delta)}$$

where $\hat{p_*} = s_*/n_*$ represent the observed number of successes divided by the number of participant in that group. The $\sigma_{mn}(\delta)$ is a function of the delta values and is create with the following equation" $\tilde{p_*}$ represent the MLE of the proportions.

$$\sigma_{mn}(\delta) = \sqrt{\left[\frac{\tilde{p_y}(1-\tilde{p_y})}{n_x} + \frac{\tilde{p_x}(1-\tilde{p_x})}{n_y}\right]\left(\frac{N}{N-1}\right)}$$

$\tilde{p_x} = 2p \cdot cos(a) - \frac{L_2}{3L_3}$ and $\tilde{p_y} = \tilde{p_x} + \delta$ where:

- $p = \pm\sqrt{\frac{L_2^2}{(3L_3)^2} - \frac{L_1}{3L_3}}$
- $a = 1/3[\pi + cos^{-1}(q/p^3)]$
- $q = \frac{L_2^3}{(3L_3)^3} - \frac{L_1 L_2}{6L_3^2} + \frac{L_0}{2L_3}$
- $L_3 = n_x + n_y$
- $L_2 = (n_x + 2n_y)\delta - N - (s_x + s_y)$
- $L_1 = (n_y\delta - L_3 - 2s_y)\delta + s_x + s_y$
- $L_0 = s_y\delta(1 - \delta)$

For more information about these equations see Miettinen (1985)

Value

An object containing the following components:

`estimate`	The point estimate of the difference in proportions (p_x - p_y)
`conf.low`	Lower bound of the confidence interval
`conf.high`	Upper bound of the confidence interval
`conf.level`	The confidence level used
`delta`	delta value(s) used
`statistic`	Z-Statistic under the null hypothesis based on the given 'delta'
`p.value`	p-value under the null hypothesis based on the given 'delta'
`method`	Description of the method used ("Miettinen-Nurminen Confidence Interval")

If `delta` is not provided statistic and p.value will be NULL

References

Miettinen, O. S., & Nurminen, M. (1985). Comparative analysis of two rates. Statistics in Medicine, 4(2), 213-226.

Examples

```
# Generate binary samples
responses <- expand(c(9, 3), c(10, 10))
arm <- rep(c("treat", "control"), times = c(10, 10))

# Calculate 95% confidence interval for difference in proportions
ci_prop_diff_mn(x = responses, by = arm)

# Calculate 99% confidence interval
ci_prop_diff_mn(x = responses, by = arm, conf.level = 0.99)

# Calculate the p-value under the null hypothesis delta = -0.1
ci_prop_diff_mn(x = responses, by = arm, delta = -0.1)

# Calculate from a data.frame
data <- data.frame(responses, arm)
ci_prop_diff_mn(x = responses, by = arm, data = data)
```

`ci_prop_diff_mn_strata` *Stratified Miettinen-Nurminen Confidence Interval for Difference in Proportions*

Description

Calculates Stratified Miettinen-Nurminen (MN) confidence intervals and corresponding point estimates for the difference between two proportions

Usage

```
ci_prop_diff_mn_strata(
  x,
  by,
  strata,
  method = c("score", "summary score"),
  conf.level = 0.95,
  delta = NULL,
  data = NULL
)
```

Arguments

`x`	(`binary/numeric/logical`) vector of a binary values, i.e. a logical vector, or numeric with values `c(0, 1)`
`by`	(`string`) A character or factor vector with exactly two unique levels identifying the two groups to compare. Can also be a column name if a data frame provided in the `data` argument.
`strata`	(`numeric`) A vector specifying the stratum for each observation. It needs to be the length of x or a multiple of x if multiple levels of strata are present. Can also be a column name (or vector of column names NOT quoted) if a data frame provided in the `data` argument.
`method`	(`string`) Specifying how the CIs should be calculated. It must equal either 'score' or 'summary score'. See details for more information about the implementation differences.
`conf.level`	(`scalar numeric`) a scalar in (0,1) indicating the confidence level. Default is 0.95
`delta`	(`numeric`) Optionally a single number or a vector of numbers between -1 and 1 (not inclusive) to set the difference between two groups under the null hypothesis. If provided, the function returns the test statistic and p-value under the `delta` hypothesis.

`data` (`data.frame`)
Optional data frame containing the variables specified in `x` and `by`.

Details

The function implements the stratified Miettinen-Nurminen method to compute confidence intervals for the difference between two proportions across multiple strata.

$$H_0 : \hat{d} - \delta <= 0 \qquad \text{vs.} \qquad H_1 : \hat{d} - \delta > 0$$

The "score" method is a weighted MN score first described in the original 1985 paper. The formula is:

- Calculates weights for each stratum as $w_i = \frac{n_{xi} \cdot n_{yi}}{n_{xi}+n_{yi}}$
- Computes the overall weighted difference $\hat{d} = \frac{\sum w_i \hat{p}_{xi}}{\sum w_i} - \frac{\sum w_i \hat{p}_{yi}}{\sum w_i}$
- Uses the stratified test statistic:

$$Z_\delta = \frac{\hat{d} - \delta}{\sqrt{\sum_{i=1}^{k} \left(\frac{w_i}{\sum w_i}\right)^2 \cdot \hat{\sigma}^2_{mn}(d)}}$$

- Finds the range of all values of δ for which the stratified test statistic (Z_δ) falls in the acceptance region $\{Z_\delta < z_{\alpha/2}\}$

The $\hat{\sigma}^2_{mn}(\hat{d})$ is the Miettinen-Nurminen variance estimate. See the details of `ci_prop_diff_mn()` for how $\hat{\sigma}^2_{mn}(\delta)$ is calculated.

The "summary score" method follows the meta-analyses proposed in Agresti 2013 and is consistent with the "Summary Score Confidence Limits" method used in SAS. The formula is:

- The point estimate of the stratified risk difference is a weighted average of the midpoints of the within-stratum MN confidence intervals:

$$\hat{d}_\text{S} = \sum_i \hat{d}_i w_i$$

- Define s_i as the width of the CI for the ith stratum divided by $2 \times z_{\alpha/2}$ and then stratum weights are given by

$$w_i = \left(\frac{1}{s_i^2}\right) \Big/ \sum_i \left(\frac{1}{s_i^2}\right)$$

- The variance of $\hat{d}_\text{S}$ is computed as

$$\widehat{\text{Var}}(\hat{d}_\text{S}) = \frac{1}{\sum_i \left(\frac{1}{s_i^2}\right)}$$

- Confidence limits for the stratified risk difference estimate are

$$\hat{d}_\text{S} \pm \left(z_{\alpha/2} \times \widehat{\text{Var}}(\hat{d}_\text{S})\right)$$

Value

An object containing the following components:

`estimate`	The point estimate of the difference in proportions (p_x - p_y)
`conf.low`	Lower bound of the confidence interval
`conf.high`	Upper bound of the confidence interval
`conf.level`	The confidence level used
`delta`	delta value(s) used
`statistic`	Z-Statistic under the null hypothesis based on the given 'delta'
`p.value`	p-value under the null hypothesis based on the given 'delta'
`method`	Description of the method used ("Stratified {method} Miettinen-Nurminen Confidence Interval")

If `delta` is not provided statistic and p.value will be NULL

References

Miettinen, O. S., & Nurminen, M. (1985). Comparative analysis of two rates. Statistics in Medicine, 4(2), 213-226.

Common Risk Difference :: Base SAS(R) 9.4 Procedures Guide: Statistical Procedures, Third Edition

Agresti, A. (2013). Categorical Data Analysis. 3rd Edition. John Wiley & Sons, Hoboken, NJ

Examples

```
# Generate binary samples with strata
responses <- expand(c(9, 3, 7, 2), c(10, 10, 10, 10))
arm <- rep(c("treat", "control"), 20)
strata <- rep(c("stratum1", "stratum2"), times = c(20, 20))

# Calculate stratified confidence interval for difference in proportions
ci_prop_diff_mn_strata(x = responses, by = arm, strata = strata)

# Using the summary score method
ci_prop_diff_mn_strata(x = responses, by = arm, strata = strata,
                       method = "summary score")

# Calculate 99% confidence interval
ci_prop_diff_mn_strata(x = responses, by = arm, strata = strata,
                       conf.level = 0.99)

# Calculate p-value under null hypothesis delta = 0.2
ci_prop_diff_mn_strata(x = responses, by = arm, strata = strata,
                       delta = 0.2)
```

`ci_prop_diff_wald` *Wald Confidence Interval for Difference in Proportions*

Description

Calculates the Wald interval by following the usual textbook definition for a difference in proportions confidence interval using the normal approximation.

Usage

```
ci_prop_diff_wald(x, by, conf.level = 0.95, correct = FALSE, data = NULL)
```

Arguments

`x`	(`binary/numeric/logical`) vector of a binary values, i.e. a logical vector, or numeric with values `c(0, 1)`
`by`	(`string`) A character or factor vector with exactly two unique levels identifying the two groups to compare. Can also be a column name if a data frame provided in the `data` argument.
`conf.level`	(`scalar numeric`) a scalar in (0,1) indicating the confidence level. Default is 0.95
`correct`	(`logical`) apply continuity correction.
`data`	(`data.frame`) Optional data frame containing the variables specified in `x` and `by`.

Details

$$(\hat{p}_1 - \hat{p}_2) \pm z_{\alpha/2}\sqrt{\frac{\hat{p}_1(1-\hat{p}_1)}{n_1} + \frac{\hat{p}_2(1-\hat{p}_2)}{n_2}}$$

Value

An object containing the following components:

`n`	Number of responses in each by group
`N`	Total number in each by group
`estimate`	The point estimate of the difference in proportions (p_1 - p_2)
`conf.low`	Lower bound of the confidence interval
`conf.high`	Upper bound of the confidence interval
`conf.level`	The confidence level used
`method`	Type of method used

Examples

```
responses <- expand(c(9, 3), c(10, 10))
arm <- rep(c("treat", "control"), times = c(10, 10))

# Calculate 95% confidence interval for difference in proportions
ci_prop_diff_wald(x = responses, by = arm)
```

`ci_prop_jeffreys` *Jeffreys CI*

Description

Calculates the Jeffreys interval, an equal-tailed interval based on the non-informative Jeffreys prior for a binomial proportion.

Usage

```
ci_prop_jeffreys(x, conf.level = 0.95, data = NULL)
```

Arguments

`x`	(`binary/numeric/logical`) vector of a binary values, i.e. a logical vector, or numeric with values `c(0, 1)`
`conf.level`	(`scalar numeric`) a scalar in (0,1) indicating the confidence level. Default is 0.95
`data`	(`data.frame`) Optional data frame containing the variables specified in `x` and `by`.

Details

$$\left(\text{Beta}\left(\frac{k}{2}+\frac{1}{2},\frac{n-k}{2}+\frac{1}{2}\right)_{\alpha},\text{Beta}\left(\frac{k}{2}+\frac{1}{2},\frac{n-k}{2}+\frac{1}{2}\right)_{1-\alpha}\right)$$

Value

An object containing the following components:

`n`	Number of responses
`N`	Total number
`estimate`	The point estimate of the proportion
`conf.low`	Lower bound of the confidence interval
`conf.high`	Upper bound of the confidence interval
`conf.level`	The confidence level used
`method`	Type of method used

`ci_prop_wald` *Wald CI*

Description

Calculates the Wald interval by following the usual textbook definition for a single proportion confidence interval using the normal approximation.

Usage

```
ci_prop_wald(x, conf.level = 0.95, correct = FALSE, data = NULL)
```

Arguments

`x`	(`binary/numeric/logical`) vector of a binary values, i.e. a logical vector, or numeric with values `c(0, 1)`
`conf.level`	(`scalar numeric`) a scalar in (0,1) indicating the confidence level. Default is 0.95
`correct`	(`logical`) apply continuity correction.
`data`	(`data.frame`) Optional data frame containing the variables specified in `x` and `by`.

Details

$$\hat{p} \pm z_{\alpha/2}\sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$$

Value

An object containing the following components:

`n`	Number of responses
`N`	Total number
`estimate`	The point estimate of the proportion
`conf.low`	Lower bound of the confidence interval
`conf.high`	Upper bound of the confidence interval
`conf.level`	The confidence level used
`method`	Type of method used

Examples

```
# example code
x <- c(
TRUE, TRUE, TRUE, TRUE, TRUE,
FALSE, FALSE, FALSE, FALSE, FALSE
)

ci_prop_wald(x, conf.level = 0.9)
```

`ci_prop_wilson` *Wilson CI*

Description

Calculates the Wilson interval by calling `stats::prop.test()`. Also referred to as Wilson score interval.

Usage

```
ci_prop_wilson(x, conf.level = 0.95, correct = FALSE, data = NULL)
```

Arguments

`x`	(`binary/numeric/logical`) vector of a binary values, i.e. a logical vector, or numeric with values `c(0, 1)`
`conf.level`	(`scalar numeric`) a scalar in (0,1) indicating the confidence level. Default is 0.95
`correct`	(`logical`) apply continuity correction.
`data`	(`data.frame`) Optional data frame containing the variables specified in `x` and `by`.

Details

$$\frac{\hat{p} + \frac{z^2_{\alpha/2}}{2n} \pm z_{\alpha/2}\sqrt{\frac{\hat{p}(1-\hat{p})}{n} + \frac{z^2_{\alpha/2}}{4n^2}}}{1 + \frac{z^2_{\alpha/2}}{n}}$$

Value

An object containing the following components:

`n`	Number of responses
`N`	Total number
`estimate`	The point estimate of the proportion

conf.low	Lower bound of the confidence interval
conf.high	Upper bound of the confidence interval
conf.level	The confidence level used
method	Type of method used

`ci_prop_wilson_strata` *Stratified Wilson CI*

Description

Calculates the stratified Wilson confidence interval for unequal proportions as described in Xin YA, Su XG. Stratified Wilson and Newcombe confidence intervals for multiple binomial proportions. *Statistics in Biopharmaceutical Research*. 2010;2(3).

Usage

```
ci_prop_wilson_strata(
  x,
  strata,
  weights = NULL,
  conf.level = 0.95,
  max.iterations = 10L,
  correct = FALSE,
  data = NULL
)
```

Arguments

x	(binary/numeric/logical) vector of a binary values, i.e. a logical vector, or numeric with values `c(0, 1)`
strata	(numeric) A vector specifying the stratum for each observation. It needs to be the length of x or a multiple of x if multiple levels of strata are present. Can also be a column name (or vector of column names NOT quoted) if a data frame provided in the `data` argument.
weights	(numeric) weights for each level of the strata. If `NULL`, they are estimated using the iterative algorithm that minimizes the weighted squared length of the confidence interval.
conf.level	(scalar numeric) a scalar in (0,1) indicating the confidence level. Default is 0.95
max.iterations	(positive integer) maximum number of iterations for the iterative procedure used to find estimates of optimal weights.

correct (scalar logical)
include the continuity correction. For further information, see for example stats::prop.test().

data (data.frame)
Optional data frame containing the variables specified in x and by.

Details

$$\frac{\hat{p}_j + \frac{z^2_{\alpha/2}}{2n_j} \pm z_{\alpha/2}\sqrt{\frac{\hat{p}_j(1-\hat{p}_j)}{n_j} + \frac{z^2_{\alpha/2}}{4n_j^2}}}{1 + \frac{z^2_{\alpha/2}}{n_j}}$$

Value

An object containing the following components:

n Number of responses

N Total number

estimate The point estimate of the proportion

conf.low Lower bound of the confidence interval

conf.high Upper bound of the confidence interval

conf.level The confidence level used

weights Weights of each strata, will be the same as the input unless unspecified, then it will be the dynamically calculated weights.

method Type of method used

Examples

```
# Stratified Wilson confidence interval with unequal probabilities

set.seed(1)
rsp <- sample(c(TRUE, FALSE), 100, TRUE)
strata_data <- data.frame(
  x = sample(c(TRUE, FALSE), 100, TRUE),
  "f1" = sample(c("a", "b"), 100, TRUE),
  "f2" = sample(c("x", "y", "z"), 100, TRUE),
  stringsAsFactors = TRUE
)
strata <- interaction(strata_data)
n_strata <- ncol(table(rsp, strata)) # Number of strata

ci_prop_wilson_strata(
  x = rsp, strata = strata,
  conf.level = 0.90
)

# Not automatic setting of weights
```

```
ci_prop_wilson_strata(
  x = rsp, strata = strata,
  weights = rep(1 / n_strata, n_strata),
  conf.level = 0.90
)
```

---

expand *Expand Count Data into Binary Vectors*

Description

Converts count data (number of successes and total sample size) into a binary vector of TRUE/FALSE values. This is useful for converting summary statistics back into raw data format for analysis functions that require individual-level data.

Usage

```
expand(x, n)
```

Arguments

x	Integer (or vector of integers) representing the number of successes.
n	Integer (or vector of integers) representing the total number of participants.

Details

For each pair of values in x and n, the function creates a vector with x TRUE values followed by n-x FALSE values. If multiple pairs are provided, the resulting vectors are concatenated in order.

Value

A logical vector where TRUE represents a success and FALSE represents a failure. The length of the vector equals the sum of all sample sizes.

Examples

```
# Convert 4 successes out of 13 participants to binary data
expand(4, 13)

# Convert multiple groups of data
# Group 1: 9 successes out of 10
# Group 2: 3 successes out of 10
expand(c(9, 3), c(10, 10))
```

Index