Package ‘IQTIGpvci’

March 6, 2018

Title R Functions for Hospital Profiling

Version 1.0.0

Maintainer IQTIG Verfahrenssupport <verfahrenssupport@iqtig.org>

Copyright IQTIG 2018

Date 2018-02-28

Description Statistical classifiers and helper functions as they are used for example in the plan. QI procedure for German hospital profiling. The package serves as an illustrative implementation in order to increase transparency about the statistical methods used in plan. QI. For official details about the plan. QI procedure (in German), see the directive (available at <https://www.g-ba.de/informationen/richtlinien/91/>) and the final development report by the IQTIG (available at <https://iqtig.org/qs-instrumente/planungsrelevante-qualitaetsindikatoren/>). The directive that governs the plan. QI project may be amended and modified each year. For current information on the project, we refer to the homepages of the Federal Joint Committee (<https://www.g-ba.de>) and the IQTIG (<https://www.iqtig.org>).

URL https://www.iqtig.org/das-iqtig/grundlagen/biometrische-methoden/

Depends R (>= 3.3.0)

License GPL-3 | file LICENSE

Encoding UTF-8

ByteCompile true

LazyData true

Imports stats

Suggests exactci (>= 1.3-3),
testthat (>= 2.0),
knitr,
rmarkdown

VignetteBuilder knitr

RoxygenNote 6.0.1

R topics documented:

`compute_oe_ci` *Compute two-sided confidence intervals for Poisson parameters.*

Description

Compute two-sided confidence intervals for the parameter λ, under the assumption that the underlying distribution of o is Poisson with parameter $e \cdot \lambda$.

Usage

```
compute_oe_ci(o, e, conf.level = 0.9, midp = TRUE, value_if_e0 = NA_real_,
  warn_if_e0 = TRUE)
```

Arguments

`o`	the observed number of events (a vector of non-negative integers)
`e`	the expected number of events (a vector of non-negative numbers)
`conf.level`	the confidence level (a vector of numbers between 0 and 1, with default value 0.90)
`midp`	a vector of logicals. If `TRUE` (the default), then compute mid-p confidence intervals instead of exact confidence intervals.
`value_if_e0`	This parameter determines the confidence interval for the case `e = 0`. If `value_if_e0` consists of a single numerical value (default: `NA_real_`), both boundaries of the confidence interval are set to this value. If `value_if_e0` is a vector of two values, these values are taken as lower and upper boundary of the confidence interval. If `NULL`, the interval is $[0, \infty]$.
`warn_if_e0`	logical of length one. If `TRUE` (the default), then print a warning if `e = 0` occurs in the input.

Details

The confidence interval at confidence level `conf.level` is computed by inverting two one-sided tests, each with significance level (1 - `conf.level`)/2 (see `compute_oe_pvalue` for details).

The case $e = 0$ deserves particular attention. In this degenerated case, the parameter λ and the confidence interval become meaningless. Therefore, the default behaviour is to assign a value of `NA_real_` to both boundaries of the confidence interval and to print a warning. The arguments `value_if_e0` and `warn_if_e0` can be used to change this behaviour.

The arguments `o`, `e`, `conf.level` and `midp` may have arbitrary lengths. If the lengths differ, arguments are recycled according to the usual R rules (warnings may appear if lenghts are not multiples of each other). If any of these arguments has length zero, the result has zero rows. If any of these arguments is `NA`, the result will also be `NA`.

Value

A `data.frame` with two columns, the `lower` and the `upper` bound of the confidence interval.

See Also

Other confidence interval functions: `compute_rate_ci`

Other functions for o/e indicators: `compute_oe_pvalue`

Examples

```
compute_oe_ci(o = 2, e = 1.8)
```

---

compute_oe_pvalue — *Compute the p-values of exact one-sided Poisson tests.*

Description

Compute the p-values of exact one-sided Poisson tests. Suppose that the underlying distribution of o is Poisson with parameter $e \cdot \lambda$. If `alternative = "greater"`, then null hypothesis and alternative hypothesis are given as

$$H_0 : \lambda \leq t_{\text{SMR}} \quad \text{and} \quad H_1 : \lambda > t_{\text{SMR}}.$$

Conversely, if `alternative = "less"`, then null hypothesis and alternative hypothesis are given as

$$H_0 : \lambda \geq t_{\text{SMR}} \quad \text{and} \quad H_1 : \lambda < t_{\text{SMR}}.$$

Usage

```
compute_oe_pvalue(o, e, t_smr, alternative = "greater", midp = TRUE,
  value_if_e0 = NA_real_, warn_if_e0 = TRUE)
```

Arguments

`o`	the observed number of events (a vector of non-negative integers)
`e`	the expected number of events (a vector of non-negative numbers)
`t_smr`	the threshold value of the null hypothesis (a vector of non-negative numbers)
`alternative`	direction of the alternative (a character vector with elements `"greater"` for $H_1 : \lambda > t_{\text{SMR}}$ or `"less"` for $H_1 : \lambda < t_{\text{SMR}}$). The default value is `"greater"`.
`midp`	a vector of logicals. If `TRUE` (the default), then compute mid-p values instead of exact p-values
`value_if_e0`	The (mid)pvalue for the case `e = 0` (a numeric of length one or `NULL`). The default value is `NA_real_`. If equal to `NULL`, then the p-value is equal to 1 and the mid-p-value is equal to 0.5 when $o = e = 0$; otherwise, when $o > e = 0$, the (mid)p-value is 0.
`warn_if_e0`	logical of length one. If `TRUE` (the default), then print a warning if `e = 0` occurs in the input.

Details

When `alternative = "greater"`, the p-value is defined as the probability $\Pr(X \geq o)$, where X is a Poisson random variable with expected value $e \cdot \lambda$. The corresponding mid-p-value is defined as

$$\Pr(X > o) + \frac{1}{2}\Pr(X = o).$$

The case $e = 0$ deserves particular attention. In this case, o should also be zero; when $e = 0$ and $o > 0$, then the null hypothesis can be immediately rejected, which corresponds to a (mid)p-value of zero. When $o = e = 0$, then, formally, the p-value is 1, and the mid-p-value is 0.5. However, when $e = 0$, this usually means that there is no data available to distinguish between the two hypotheses, and so the (mid)p-value is meaningless. Therefore, the default behaviour is to assign a value of `NA_real_` to this case and to print a warning. The arguments `value_if_e0` and `warn_if_e0` can be used to change this behaviour.

The arguments `o`, `e`, `t_smr` and `alternative` may have arbitrary lengths. If the lengths differ, arguments are recycled according to the usual R rules (warnings may appear if lenghts are not multiples of each other). If any of these arguments has length zero, the result also has length zero. If any of these arguments is `NA`, the result will also be `NA`.

Value

A vector of (mid)p-values.

See Also

Other p-value functions: `compute_rate_pvalue`

Other functions for o/e indicators: `compute_oe_ci`

Examples

```
compute_oe_pvalue(o = 2, e = 1.8, t_smr = 2.18)
```

`compute_rate_ci` *Compute two-sided confidence intervals for binomial parameters.*

Description

Compute two-sided confidence intervals for a binomial parameter p, under the assumption that the underlying distribution of o is binomial with parameters p and n.

Usage

```
compute_rate_ci(o, n, conf.level = 0.9, midp = TRUE,
  value_if_n0 = NA_real_, warn_if_n0 = TRUE)
```

Arguments

o	the observed number of events (a vector of non-negative integers)
n	total number of cases (a vector of non-negative integers that satisfies `o <= n`)
conf.level	the confidence level (a vector of numbers between 0 and 1, with default value 0.90)
midp	a vector of logicals. If `TRUE` (the default), then compute mid-p confidence intervals instead of exact confidence intervals.
value_if_n0	This parameter determines the confidence interval for the case `o = n = 0`. If `value_if_n0` consists of a single numerical value (default: `NA_real_`), both boundaries of the confidence interval are set to this value. If `value_if_n0` is a vector of two values, these values are taken as lower and upper boundary of the confidence interval. If `NULL`, the interval is [0, 1].
warn_if_n0	logical of length one. If `TRUE` (the default), then print a warning if the combination `o = n = 0` occurs in the input.

Details

The confidence interval at confidence level `conf.level` is computed by inverting two one-sided tests, each with significance level (1 - `conf.level`)/2 (see `compute_rate_pvalue` for details).

The case $o = n = 0$ deserves particular attention. Formally, the confidence interval is [0, 1] in this case. However, when $n = 0$, then there is no data available to estimate the binomial parameter, and so the confidence interval is meaningless. Therefore, the default behaviour is to assign a value of `NA_real_` to both boundaries of the confidence interval and to print a warning. The arguments `value_if_n0` and `warn_if_n0` can be used to change this behaviour.

The arguments `o`, `n`, `conf.level` and `midp` may have arbitrary lengths. If the lengths differ, arguments are recycled according to the usual R rules (warnings may appear if lenghts are not multiples of each other). If any of these arguments has length zero, the result has zero rows. If any of these argument is `NA`, the result will also be `NA`.

Value

A `data.frame` with two columns, the `lower` and the `upper` bound of the confidence interval.

See Also

Other confidence interval functions: `compute_oe_ci`

Other functions for rate indicators: `compute_rate_pvalue`

Examples

```
compute_rate_ci(o = 1, n = 2)
```

compute_rate_pvalue *Compute (mid)p-values of exact one-sided binomial tests.*

Description

Compute (mid)p-values of exact one-sided binomial tests. If `alternative = "greater"`, then null hypothesis and alternative hypothesis are given as

$$H_0 : p \leq t \quad \text{and} \quad H_1 : p > t.$$

Conversely, if `alternative = "less"`, then null hypothesis and the alternative hypothesis are given as

$$H_0 : p \geq t \quad \text{and} \quad H_1 : p < t.$$

Usage

```
compute_rate_pvalue(o, n, t, alternative = "greater", midp = TRUE,
  value_if_n0 = NA_real_, warn_if_n0 = TRUE)
```

Arguments

`o`	the observed number of events (a vector of non-negative integers)
`n`	total number of cases (a vector of non-negative integers that satisfies `o <= n`)
`t`	the threshold value of the null hypothesis (a vector of numbers between 0 and 1)
`alternative`	direction of the alternative (a character vector with elements `"greater"` for $H_1 : p > t$ or `"less"` for $H_1 : p < t$). The default value is `"greater"`.
`midp`	a vector of logicals. If `TRUE` (the default), then compute mid-p values instead of exact p-values.
`value_if_n0`	The (mid-)p-value for the case `o = n = 0` (a numeric of length one or `NULL`). The default value is `NA_real_`. If equal to `NULL`, then the p-value is equal to 1 and the mid-p-value is equal to 0.5.
`warn_if_n0`	logical of length one. If `TRUE` (the default), then print a warning if the combination `o = n = 0` occurs in the input.

Details

When `alternative = "greater"`, the p-value is defined as the probability $\Pr(X \geq o)$, where X is a binomial random variable with n trials and success probability t. The corresponding mid-p-value is defined as

$$\Pr(X > o) + \frac{1}{2}\Pr(X = o).$$

The case $o = n = 0$ deserves particular attention. Formally, the p-value is 1, and the mid-p-value is 0.5 in this case. However, when $n = 0$, then there is no data available to distinguish between the two hypotheses, and so the (mid)p-value is meaningless. Therefore, the default behaviour is to assign a value of `NA_real_` to this case and to print a warning. The arguments `value_if_n0` and `warn_if_n0` can be used to change this behaviour.

The arguments `o`, `n`, `t` and `alternative` may have arbitrary lengths. If the lengths differ, arguments are recycled according to the usual R rules (warnings may appear if lenghts are not multiples of each other). If any of these argument has length zero, the result also has length zero. If any of these argument is `NA`, the result will also be `NA`.

Value

A vector of (mid)p-values.

See Also

Other p-value functions: `compute_oe_pvalue`

Other functions for rate indicators: `compute_rate_ci`

Examples

```
compute_rate_pvalue(o = 1, n = 2, t = 0.95, alternative = "less")
```

Index