Package ‘blindrecalc’

July 6, 2021

Type Package

Title Blinded Sample Size Recalculation

Version 0.1.3


License MIT + file LICENSE

URL https://github.com/imbi-heidelberg/blindrecalc

Encoding UTF-8

Suggests testthat, covr

Imports methods, Rcpp


RoxygenNote 7.1.1

LinkingTo Rcpp

NeedsCompilation yes

Author Lukas Baumann [aut, cre] (<https://orcid.org/0000-0001-7931-7470>), Maximilian Pilz [aut] (<https://orcid.org/0000-0002-9685-1613>), Institute of Medical Biometry and Informatics - University of Heidelberg [cph]

Maintainer Lukas Baumann <baumann@imbi.uni-heidelberg.de>

Repository CRAN

Date/Publication 2021-07-06 16:00:16 UTC
adjusted_alpha

**Adjustment of significance levels**

This method takes an adjusted level of significance to ensure that the actual type I error rate is preserved.

**Description**

This method returns an adjusted significance level that can be used such that the actual type I error rate is preserved.

**Usage**

```r
adjusted_alpha(design, n1, nuisance, ...)  
```

**Arguments**

- `design` object of class `TestStatistic` created by `setup`
- `n1` total number of patients that are recruited before the sample size is recalculated
- `nuisance` nuisance parameter that is estimated at the interim analysis
- `...` Further optional arguments.
adjusted_alpha,ChiSquare-method

Details

The method is only vectorized in either nuisance or n1.

The method is implemented for the classes Student, ChiSquare, and FarringtonManning. Check the class-specific documentation for further parameters that have to be specified.

Value

Value of the adjusted significance level for every nuisance parameter and every value of n1.

Examples

d <- setupStudent(alpha = .025, beta = .2, r = 1, delta = 0, delta_NI = 1.5, n_max = 848)
sigma <- c(2, 5.5, 9)
adjusted_alpha(design = d, n1 = 20, nuisance = sigma, tol = 1e-4, iters = 1e3)

adjusted_alpha,ChiSquare-method

Adjusted level of significance

Description

This method returns an adjusted significance level that can be used such that the actual type I error rate is preserved.

Usage

## S4 method for signature 'ChiSquare'
adjusted_alpha(
    design,
    n1,
    nuisance,
    nuis_ass,
    precision = 0.001,
    gamma = 0,
    recalculation,
    allocation = c("exact", "approximate"),
    ...
)

Arguments

design Object of class ChiSquare created by setupChiSquare.
n1 Either the sample size of the first stage (if recalculation = TRUE or the total sample size (if recalculation = FALSE).
nuisance Value of the nuisance parameter in (0,1). For the Chi-Squared test this is the overall response rate.
nuis_ass If recalculation = FALSE this is the value for the overall response rate that is used to calculate the sample size for the adjusted significance level.

precision Value by which the nominal type 1 error rate is reduced in each iteration until the nominal type 1 error rate is preserved.

gamma If gamma > 0 then the significance level is adjusted such that the actual level is at most alpha - gamma. This is necessary to maintain the nominal significance level if a confidence interval approach proposed by Friede & Kieser (2011) is used.

recalculation Should the sample size be recalculated after n1 patients are recruited?

allocation Whether the allocation ratio should be preserved exactly (exact) or approximately (approximate or kf_approx). approximate uses the unrounded calculated sample size in the sample size recalculation, kf_approx rounds the sample size to the next integer.

... Further optional arguments.

Details
The method is only vectorized in either nuisance or n1.

Value
Value of the adjusted significance level for every nuisance parameter and every value of n1.

Examples
d <- setupChiSquare(alpha = 0.025, beta = 0.2, r = 1, delta = 0.2)
adjusted_alpha(d, n1 = 10, nuisance = 0.3, gamma = 0.001, nuis_ass = 0.3, precision = 0.001, recalculation = TRUE)
adjusted_alpha.FarringtonManning-method

nuis_ass,
precision = 0.001,
gamma = 0,
recalculation,
allocation = c("exact", "approximate"),
...
)

Arguments

design Object of class FarringtonManning created by setupFarringtonManning.
n1 Either the sample size of the first stage (if recalculation = TRUE) or the total
sample size (if recalculation = FALSE).
nuisance Value of the nuisance parameter in (0,1). For the Farrington-Manning test this
is the overall response rate.
nuis_ass If recalculation = FALSE this is the value for the overall response rate that is
used to calculate the sample size for the adjusted significance level.
precision Value by which the nominal type 1 error rate is reduced in each iteration until
the nominal type 1 error rate is preserved.
gamma If gamma > 0 then the significance level is adjusted such that the actual level is
at most alpha -gamma. This is necessary to maintain the nominal significance
level if a confidence interval approach proposed by Friede & Kieser (2011) is
used.
recalculation Should the sample size be recalculated after n1 n1 patients are recruited?
allocation Whether the allocation ratio should be preserved exactly (exact) or approxi-
mately (approximate).
...
Further optional arguments.

Details
The method is only vectorized in either nuisance or n1.

Value
Value of the adjusted significance level for every nuisance parameter and every value of n1.

Examples
d <- setupFarringtonManning(alpha = 0.025, beta = 0.2, r = 1, delta = 0, delta_NI = 0.25)
adjusted_alpha(d, n1 = 20, nuisance = 0.5, recalculation = TRUE)
adjusted_alpha,Student-method

Adjusted level of significance

Description

This method returns an adjusted significance level that can be used such that the actual type I error rate is preserved.

Usage

```r
## S4 method for signature 'Student'
adjusted_alpha(design, n1, nuisance, tol, iters = 10000, seed = NULL, ...)
```

Arguments

- `design`: Object of class `Student` created by `setupStudent`.
- `n1`: Either the sample size of the first stage (if `recalculation = TRUE`) or the total sample size (if `recalculation = FALSE`).
- `nuisance`: Value of the nuisance parameter. For the Student's t-test this is the variance.
- `tol`: desired absolute tolerance
- `iters`: Number of simulation iterations.
- `seed`: Random seed for simulation.
- `...`: Further optional arguments.

Details

The method is only vectorized in either `nuisance` or `n1`.

In the case of the Student's t-test, the adjusted alpha is calculated using the algorithm by Kieser and Friede (2000): "Re-calculating the sample size in internal pilot study designs with control of the type I error rate". Statistics in Medicine 19: 901-911.

Value

Value of the adjusted significance level for every nuisance parameter and every value of `n1`.

Examples

```r
d <- setupStudent(alpha = .025, beta = .2, r = 1, delta = 0, delta_NI = 1.5, n_max = 848)
sigma <- c(2, 5.5, 9)
adjusted_alpha(design = d, n1 = 20, nuisance = sigma, tol = 1e-4, iters = 1e3)
```
blindreclac  

**Blinded Sample Size Recalculation**

**Description**

The package **blindreclac** provides characteristics and plots of trial designs with blinded sample size recalculation where a nuisance parameter is estimated at a blinded interim analysis.

**Details**

Currently, for continuous outcomes, a t-test is implemented for superiority and non-inferiority trials. For superiority trials with binary endpoint, the chi^2-test is implemented. The Farrington Manning test covers non-inferiority trials with binary endpoint.

---

**ChiSquare-class**  

**Chi-squared test**

**Description**

This class implements a chi-squared test for superiority trials. A trial with binary outcomes in two groups E and C is assumed. If alternative == "greater" the null and alternative hypotheses for the difference in response probabilities are

\[ H_0 : p_E \leq p_C \text{ vs. } H_1 : p_E > p_C. \]

If alternative == "smaller", the direction of the effect is changed.

The function `setupChiSquare` creates an object of class **ChiSquare**.

**Usage**

```r
setupChiSquare(
  alpha,
  beta,
  r = 1,
  delta,
  alternative = c("greater", "smaller"),
  n_max = Inf,
  ...
)
```
### ChiSquare-class

#### Arguments

- **alpha**: One-sided type I error rate.
- **beta**: Type II error rate.
- **r**: Allocation ratio between experimental and control group.
- **delta**: Difference of effect size between alternative and null hypothesis.
- **alternative**: Does the alternative hypothesis contain greater (greater) or smaller (smaller) values than the null hypothesis.
- **n_max**: Maximal overall sample size. If the recalculated sample size is greater than \( n_{\text{max}} \) it is set to \( n_{\text{max}} \).
- ...: Further optional arguments.

#### Details

The nuisance parameter is the overall response probability \( p_0 \). In the blinded sample size recalculation procedure it is blindly estimated by:

\[
\hat{p}_0 := \frac{(X_{1,E} + X_{1,C})}{(n_{1,E} + n_{1,C})},
\]

where \( X_{1,E} \) and \( X_{1,C} \) are the numbers of responses and \( n_{1,E} \) and \( n_{1,C} \) are the sample sizes of the respective group after the first stage. The event rates in both groups under the alternative hypothesis can then be blindly estimated as:

\[
\hat{p}_{C,A} := \hat{p}_0 - \Delta \cdot \frac{r}{1 + r}, \quad \hat{p}_{E,A} := \hat{p}_0 + \Delta \cdot \frac{1}{1 + r},
\]

where \( \Delta \) is the difference in response probabilities under the alternative hypothesis and \( r \) is the allocation ratio of the sample sizes in the two groups. These blinded estimates can then be used to re-estimate the sample size.

The following methods are available for this class: `toer`, `pow`, `n_dist`, `adjusted_alpha`, and `n_fix`. Check the design specific documentation for details.

For non-inferiority trials use the function `setupFarringtonManning`.

#### Value

An object of class `ChiSquare`.

#### References


#### Examples

```r
design <- setupChiSquare(alpha = .025, beta = .2, r = 1, delta = 0.2, alternative = "greater")
```
Description

This class implements a Farrington-Manning test for non-inferiority trials. A trial with binary outcomes in two groups \(E\) and \(C\) is assumed. The null and alternative hypotheses for the non-inferiority of response probabilities are:

\[
H_0 : p_E - p_C \leq -\delta \text{ vs. } H_1 : p_E - p_C > -\delta,
\]

where \(\delta\) denotes the non-inferiority margin.

The function `setupFarringtonManning` creates an object of `FarringtonManning`.

Usage

```r
setupFarringtonManning(alpha, beta, r = 1, delta, delta_NI, n_max = Inf, ...)
```

Arguments

- **alpha**: One-sided type I error rate.
- **beta**: Type II error rate.
- **r**: Allocation ratio between experimental and control group.
- **delta**: Difference of effect size between alternative and null hypothesis.
- **delta_NI**: Non-inferiority margin.
- **n_max**: Maximal overall sample size. If the recalculated sample size is greater than \(n_{max}\) it is set to \(n_{max}\).
- **...**: Further optional arguments.

Details

The nuisance parameter is the overall response probability \(p_0\). In the blinded sample size recalculation procedure it is blindly estimated by:

\[
\hat{p}_0 := (X_{1,E} + X_{1,C})/(n_{1,E} + n_{1,C}),
\]

where \(X_{1,E}\) and \(X_{1,C}\) are the numbers of responses and \(n_{1,E}\) and \(n_{1,C}\) are the sample sizes of the respective group after the first stage. The event rates in both groups under the alternative hypothesis can then be blindly estimated as:

\[
\hat{p}_{E,A} := \hat{p}_0 - \Delta \cdot r/(1 + r), \quad \hat{p}_{C,A} := \hat{p}_0 + \Delta/(1 + r),
\]

where \(\Delta\) is the difference in response probabilities under the alternative hypothesis and \(r\) is the allocation ratio of the sample sizes in the two groups. These blinded estimates can then be used to re-estimate the sample size.

The following methods are available for this class: `toer`, `pow`, `n_dist`, `adjusted_alpha`, and `n_fix`. Check the design specific documentation for details.
Value

An object of class FarringtonManning.

References


Examples

design <- setupFarringtonManning(alpha = .025, beta = .2, r = 1, delta = 0, delta_NI = .15)

n_dist

Distribution of the Sample Size

Description

Calculates the distribution of the total sample sizes of designs with blinded sample size recalculation for different values of the nuisance parameter or of n1.

Usage

n_dist(design, n1, nuisance, summary = TRUE, plot = FALSE, ...)

Arguments

design object of class TestStatistic created by setup
n1 total number of patients that are recruited before the sample size is recalculated
nuisance nuisance parameter that is estimated at the interim analysis
summary logical - is a summary of the sample size distribution desired? Otherwise, a vector with sample sizes is returned.
plot Should a plot of the sample size distribution be drawn?
... Further optional arguments.

Details

The method is only vectorized in either nuisance or n1.
The method is implemented for the classes Student, ChiSquare, and FarringtonManning.

Value

Summary and/or plot of the sample size distribution for every nuisance parameter and every value of n1.
Examples

d <- setupStudent(alpha = .025, beta = .2, r = 1, delta = 3.5, delta_NI = 0, alternative = "greater", n_max = 156)
n_dist(d, n1 = 20, nuisance = 5.5, summary = TRUE, plot = FALSE, seed = 2020)

---

n_dist,ChiSquare-method

Distribution of the Sample Size

Description

Calculates the distribution of the total sample sizes of designs with blinded sample size recalculation for different values of the nuisance parameter or of n1.

Usage

## S4 method for signature 'ChiSquare'
n_dist(
  design, 
n1, 
nuisance, 
summary = TRUE, 
plot = FALSE, 
allocation = c("exact", "approximate"), 
... 
)

Arguments

design Object of class ChiSquare created by setupChiSquare.
n1 Either the sample size of the first stage (if recalculation = TRUE or the total sample size (if recalculation = FALSE).
nuisance Value of the nuisance parameter in (0,1). For the Chi-Squared test this is the overall response rate.
summary Is a summary of the sample size distribution desired? Otherwise, a vector with sample sizes is returned.
plot Should a plot of the sample size distribution be drawn?
allocation Whether the allocation ratio should be preserved exactly (exact) or approximately (approximate or kf_approx). approximate uses the unrounded calculated sample size in the sample size recalculation, kf_approx rounds the sample size to the next integer.
... Further optional arguments.
Details

Only sample sizes that occur with a probability of at least 0.01 considered.
The method is only vectorized in either nuisance or n1.

Value

Summary and/or plot of the sample size distribution for every nuisance parameter and every value of n1.

Examples

```r
d <- setupChiSquare(alpha = 0.025, beta = 0.2, r = 1, delta = 0.2)
n_dist(d, n1 = 20, nuisance = 0.25, summary = TRUE, plot = FALSE)
```

Description

Calculates the distribution of the total sample sizes of designs with blinded sample size recalculation for different values of the nuisance parameter or of n1.

Usage

```r
## S4 method for signature 'FarringtonManning'
n_dist(
  design,  # Object of class FarringtonManning created by setupFarringtonManning.
  n1,      # Either the sample size of the first stage (if recalculation = TRUE) or the total sample size (if recalculation = FALSE).
  nuisance, # Value of the nuisance parameter in (0,1). For the Farrington-Manning test this is the overall response rate.
  summary,  # Is a summary of the sample size distribution desired? Otherwise, a vector with sample sizes is returned.
  ...)
```

Arguments

design

n1

nuisance

summary
Should a plot of the sample size distribution be drawn?

Whether the allocation ratio should be preserved exactly (exact) or approximately (approximate).

Further optional arguments.

Only sample sizes that occur with a probability of at least 0.01 considered.
The method is only vectorized in either nuisance or n1.

Summary and/or plot of the sample size distribution for each nuisance parameter and every value of n1.

Example

d <- setupFarringtonManning(alpha = 0.025, beta = 0.2, r = 1, delta = 0, delta_NI = 0.25)
n_dist(d, n1 = 30, nuisance = 0.2, summary = TRUE, plot = FALSE)

Description

Calculates the distribution of the total sample sizes of designs with blinded sample size recalculation for different values of the nuisance parameter or of n1.

Usage

```r
## S4 method for signature 'Student'
n_dist(  
design,  
n1,  
nuisance,  
summary = TRUE,  
plot = FALSE,  
iters = 10000,  
seed = NULL,  
range = 0,  
allocation = c("approximate", "exact"),  
...
)
```
**Arguments**

- **design**: Object of class Student created by setupStudent.
- **n1**: Either the sample size of the first stage (if recalculation = TRUE) or the total sample size (if recalculation = FALSE).
- **nuisance**: Value of the nuisance parameter. For the Student's t-test this is the variance.
- **summary**: Is a summary of the sample size distribution desired? Otherwise, a vector with sample sizes is returned.
- **plot**: Should a plot of the sample size distribution be drawn?
- **iters**: Number of simulation iterations.
- **seed**: Random seed for simulation.
- **range**: This determines how far the plot whiskers extend out from the box. If range is positive, the whiskers extend to the most extreme data point which is no more than range times the interquartile range from the box. A value of zero causes the whiskers to extend to the data extremes.
- **allocation**: Whether the allocation ratio should be preserved exactly (exact) or approximately (approximate).
- **...**: Further optional arguments.

**Details**

The method is only vectorized in either nuisance or n1.

**Value**

Summary and/or plot of the sample size distribution for every nuisance parameter and every value of n1.

**Examples**

```r
  d <- setupStudent(alpha = .025, beta = .2, r = 1, delta = 3.5, delta_NI = 0,
                    alternative = "greater", n_max = 156)
  n_dist(d, n1 = 20, nuisance = 5.5, summary = TRUE, plot = FALSE, seed = 2020)
```

---

**n_fix**

*Fixed Sample Size*

**Description**

Returns the total sample size of a fixed design without sample size recalculation.

**Usage**

```r
  n_fix(design, nuisance, ...)```

n_fix,ChiSquare-method

Arguments

design test statistic object created by setup
nuisance nuisance parameter for the respective test problem
... Further optional arguments.

Details

The method is only vectorized in either nuisance or n1.
The method is implemented for the classes Student, ChiSquare, and FarringtonManning.

Value

One value of the fixed sample size for every nuisance parameter and every value of n1.

Examples

d <- setupStudent(alpha = .025, beta = .2, r = 1, delta = 3.5, delta_NI = 0,
alternative = "greater", n_max = 156)
n_fix(design = d, nuisance = 5.5)
n_fix,FarringtonManning-method

variance
A character string indicating whether the "heterogenous" (default) or the "homogeneous" variance formula should be used.

rounded
Whether the calculated sample size should be rounded up such that the allocation ratio is preserved.

...
Further optional arguments.

Details
The method is only vectorized in either nuisance or n1.

Value
One value of the fixed sample size for every nuisance parameter and every value of n1.

Examples
```r
design1 <- setupChiSquare(alpha = 0.025, beta = 0.2, r = 1, delta = 0.2)
n_fix(design1, nuisance = c(0.2, 0.3))
```

n_fix,FarringtonManning-method

Fixed Sample Size

Description
Returns the sample size of a fixed design without sample size recalculation.

Usage
```r
## S4 method for signature 'FarringtonManning'
n_fix(design, nuisance, rounded = TRUE, ...)
```

Arguments
design
Object of class FarringtonManning created by setupFarringtonManning.
nuisance
Value of the nuisance parameter. For the Farrington-Manning test this is the overall response rate.
rounded
Whether the calculated sample size should be rounded up such that the allocation ratio is preserved.

Details
The method is only vectorized in either nuisance or n1.
Value

One value of the fixed sample size for every nuisance parameter and every value of n1.

Examples

d <- setupFarringtonManning(alpha = 0.025, beta = 0.2, r = 1, delta = 0, delta_NI = 0.25)
n_fix(d, nuisance = 0.3)

Description

Returns the sample size of a fixed design without sample size recalculation.

Usage

## S4 method for signature 'Student'
n_fix(design, nuisance, ...)

Arguments

design test statistic object
nuisance nuisance parameter
... Further optional arguments.

Details

The method is only vectorized in either nuisance or n1.

Value

One value of the fixed sample size for every nuisance parameter and every value of n1.

Examples

d <- setupStudent(alpha = .025, beta = .2, r = 1, delta = 3.5, delta_NI = 0, alternative = "greater", n_max = 156)
n_fix(design = d, nuisance = 5.5)
**Description**

Calculates the power of designs with blinded sample size recalculation or of fixed designs for one or several values of the nuisance parameter.

**Usage**

```r
pow(design, n1, nuisance, recalculation, ...)  
```

**Arguments**

- `design`: object of class `TestStatistic` created by `setup`
- `n1`: total number of patients that are recruited before the sample size is recalculated
- `nuisance`: nuisance parameter that is estimated at the interim analysis
- `recalculation`: Should the sample size be recalculated after `n1` patients are recruited?
- `...`: Further optional arguments.

**Details**

The method is only vectorized in either `nuisance` or `n1`.

The method is implemented for the classes `Student`, `ChiSquare`, and `FarringtonManning`.

**Value**

One power value for every nuisance parameter and every value of `n1`.

**Examples**

```r
d <- setupStudent(alpha = .025, beta = .2, r = 1, delta = 3.5, delta_NI = 0,  
alternative = "greater", n_max = 156)
pow(d, n1 = 20, nuisance = 5.5, recalculation = TRUE)
```
Description

Calculates the power of designs with blinded sample size recalculation or of fixed designs for one or several values of the nuisance parameter.

Usage

```r
## S4 method for signature 'ChiSquare'
pow(design, n1, nuisance, recalculation, allocation = c("exact", "approximate", "kf_approx"), ...)
```

Arguments

- `design`: Object of class `ChiSquare` created by `setupChiSquare`.
- `n1`: Either the sample size of the first stage (if `recalculation = TRUE`) or the total sample size (if `recalculation = FALSE`).
- `nuisance`: Value of the nuisance parameter in (0,1). For the Chi-Squared test this is the overall response rate.
- `recalculation`: Should the sample size be recalculated after `n1` patients are recruited?
- `allocation`: Whether the allocation ratio should be preserved exactly (`exact`) or approximately (`approximate` or `kf_approx`). `approximate` uses the unrounded calculated sample size in the sample size recalculation, `kf_approx` rounds the sample size to the next integer.
- `...`: Further optional arguments.

Details

The method is only vectorized in either `nuisance` or `n1`.

Value

One power value for every nuisance parameter and every value of `n1`.

Examples

```r
d <- setupChiSquare(alpha = 0.025, beta = 0.2, r = 1, delta = 0.2)
pow(d, n1 = 20, nuisance = c(0.2, 0.4), recalculation = TRUE)
```
Description

Calculates the power of designs with blinded sample size recalculation or of fixed designs for one or several values of the nuisance parameter.

Usage

```r
## S4 method for signature 'FarringtonManning'
pow(
  design,  
n1,  
nuisance,  
recalculation,  
allocation = c("exact", "approximate"),  
...  
)
```

Arguments

- `design`: Object of class `FarringtonManning` created by `setupFarringtonManning`.
- `n1`: Either the sample size of the first stage (if `recalculation = TRUE` or the total sample size (if `recalculation = FALSE`).
- `nuisance`: Value of the nuisance parameter in (0,1). For the Farrington-Manning test this is the overall response rate.
- `recalculation`: Should the sample size be recalculated after `n1` patients are recruited?
- `allocation`: Whether the allocation ratio should be preserved exactly (`exact`) or approximately (`approximate`).
- `...`: Further optional arguments.

Details

The method is only vectorized in either `nuisance` or `n1`.

Value

One power value for every nuisance parameter and every value of `n1`.

Examples

```r
d <- setupFarringtonManning(alpha = 0.025, beta = 0.2, r = 1, delta = 0, delta_NI = 0.25)
pow(d, n1 = 30, nuisance = 0.4, allocation = "approximate", recalculation = TRUE)
```
Description

Calculates the power of designs with blinded sample size recalculation or of fixed designs for one or several values of the nuisance parameter.

Usage

```r
## S4 method for signature 'Student'
pow(
  design,
  n1,
  nuisance,
  recalculation = TRUE,
  iters = 10000,
  seed = NULL,
  allocation = c("approximate", "exact"),
  ...
)
```

Arguments

- `design`: Object of class `Student` created by `setupStudent`.
- `n1`: Either the sample size of the first stage (if `recalculation = TRUE`) or the total sample size (if `recalculation = FALSE`).
- `nuisance`: Value of the nuisance parameter. For the Student's t-test this is the variance.
- `recalculation`: Should the sample size be recalculated after `n1` patients are recruited?
- `iters`: Number of simulation iterations.
- `seed`: Random seed for simulation.
- `allocation`: Whether the allocation ratio should be preserved exactly (exact) or approximately (approximate).
- `...`: Further optional arguments.

Details

The method is only vectorized in either `nuisance` or `n1`.

Value

One power value for every nuisance parameter and every value of `n1`. 

Examples

d <- setupStudent(alpha = .025, beta = .2, r = 1, delta = 3.5, delta_NI = 0, alternative = "greater", n_max = 156)
pow(d, n1 = 20, nuisance = 5.5, recalculation = TRUE)

---

**Simulation**  
*Simulate Rejection Probability and Sample Size for Student’s t-Test*

**Description**

This function simulates the probability that a test defined by `setupStudent` rejects the null hypothesis. Note that here the nuisance parameter `nuisance` is the variance of the outcome variable $\sigma^2$.

**Usage**

```r
simulation(
  design,
  n1, 
  nuisance, 
  recalculation = TRUE, 
  delta_true, 
  iters = 1000, 
  seed = NULL, 
  allocation = c("approximate", "exact"),
  ... 
)
```

**Arguments**

- `design`  
  Object of class `Student` created by `setupStudent`.
- `n1`  
  Either the sample size of the first stage (if `recalculation = TRUE`) or the total sample size (if `recalculation = FALSE`).
- `nuisance`  
  Value of the nuisance parameter. For the Student’s t-test this is the variance.
- `recalculation`  
  Should the sample size be recalculated after `n1` patients are recruited?
- `delta_true`  
  Effect measure under which the rejection probabilities are computed.
- `iters`  
  Number of simulation iterations.
- `seed`  
  Random seed for simulation.
- `allocation`  
  Whether the allocation ratio should be preserved exactly (exact) or approximately (approximate).
- `...`  
  Further optional arguments.
Details


Value

Simulated rejection probabilities and sample sizes for each nuisance parameter.

Examples

d <- setupStudent(alpha = .025, beta = .2, r = 1, delta = 3.5, delta_NI = 0, alternative = "greater", n_max = 156)
simulation(d, n1 = 20, nuisance = 5.5, recalculation = TRUE, delta_true = 3.5)

Description

This class implements Student’s t-test for superiority and non-inferiority tests. A trial with continuous outcomes of the two groups E and C is assumed. If alternative == "greater" the null hypothesis for the mean difference \( \Delta = \mu_E - \mu_C \) is

\[
H_0 : \Delta \leq -\delta_{NI} \text{ vs. } H_1 : \Delta > -\delta_{NI}.
\]

Here, \( \delta_{NI} \geq 0 \) denotes the non-inferiority margin. For superiority trials, \( \delta_{NI} \) can be set to zero (default). If alternative="smaller", the direction of the effect is changed.

The function setupStudent creates an object of class Student that can be used for sample size recalculation.

Usage

```r
setupStudent(
    alpha,
    beta,
    r = 1,
    delta,
    delta_NI = 0,
    alternative = c("greater", "smaller"),
    n_max = Inf,
    ...
)```


Arguments

alpha  One-sided type I error rate.
beta   Type II error rate.
r      Allocation ratio between experimental and control group.
delta  Difference of effect size between alternative and null hypothesis.
delta_NI Non-inferiority margin.
alternative Does the alternative hypothesis contain greater (greater) or smaller (smaller) values than the null hypothesis.
n_max  Maximal overall sample size. If the recalculated sample size is greater than n_max it is set to n_max.
...    Further optional arguments.

Details

The nuisance parameter is the variance \( \sigma^2 \). Within the blinded sample size recalculation procedure, it is re-estimated by the one-sample variance estimator that is defined by

\[
\hat{\sigma}^2 := \frac{1}{n_1 - 1} \sum_{j \in \{T,C\}} \sum_{k=1}^{n_{1,j}} (x_{j,k} - \bar{x})^2,
\]

where \( x_{j,k} \) is the outcome of patient \( k \) in group \( j \), \( n_{1,j} \) denotes the first-stage sample size in group \( j \) and \( \bar{x} \) equals the mean over all \( n_1 \) observations.

The following methods are available for this class: toer, pow, n_dist, adjusted_alpha, and n_fix. Check the design specific documentation for details.

Value

An object of class Student.

References


Examples

d <- setupStudent(alpha = .025, beta = .2, r = 1, delta = 3.5, delta_NI = 0, alternative = "greater", n_max = 156)
Description

Computes the type I error rate of designs with blinded sample size recalculation or of fixed designs for one or several values of the nuisance parameter.

Usage

toer(design, n1, nuisance, recalculation, ...)

Arguments

- **design**: object of class `TestStatistic` created by `setup`
- **n1**: total number of patients that are recruited before the sample size is recalculated
- **nuisance**: nuisance parameter that is estimated at the interim analysis
- **recalculation**: Should the sample size be recalculated after n1 patients are recruited?
- **...**: Further optional arguments.

Details

The method is only vectorized in either `nuisance` or `n1`.

The method is implemented for the classes `Student`, `ChiSquare`, and `FarringtonManning`.

Value

One type I error rate value for every nuisance parameter and every value of n1.

Examples

d <- setupStudent(alpha = 0.025, beta = 0.2, r = 1, delta = 3.5, delta_NI = 0, alternative = "greater", n_max = 156)
toer(d, n1 = 20, nuisance = 5.5, recalculation = TRUE)
**Description**

Computes the type I error rate of designs with blinded sample size recalculation or of fixed designs for one or several values of the nuisance parameter.

**Usage**

```r
## S4 method for signature 'ChiSquare'

# S4 method for signature 'ChiSquare'
toer(
  design,
  n1,
  nuisance,
  recalculation,
  allocation = c("exact", "approximate", "kf_approx"),
  ...
)
```

**Arguments**

- `design`: Object of class `ChiSquare` created by `setupChiSquare`.
- `n1`: Either the sample size of the first stage (if `recalculation = TRUE`) or the total sample size (if `recalculation = FALSE`).
- `nuisance`: Value of the nuisance parameter in (0,1). For the Chi-Squared test this is the overall response rate.
- `recalculation`: Should the sample size be recalculated after `n1` patients are recruited?
- `allocation`: Whether the allocation ratio should be preserved exactly (exact) or approximately (approximate or kf_approx). approximate uses the unrounded calculated sample size in the sample size recalculation, kf_approx rounds the sample size to the next integer.
- `...`: Further optional arguments.

**Details**

The method is only vectorized in either `nuisance` or `n1`.

**Value**

One type I error rate value for every nuisance parameter and every value of `n1`.

**Examples**

```r
d <- setupChiSquare(alpha = 0.025, beta = 0.2, r = 1, delta = 0.2)
toer(d, n1 = c(10, 20), nuisance = 0.25, recalculation = TRUE)
```
toer,FarringtonManning-method

Type I Error Rate

Description

Computes the type I error rate of designs with blinded sample size recalculation or of fixed designs for one or several values of the nuisance parameter.

Usage

```r
## S4 method for signature 'FarringtonManning'
toer(
  design,
  n1,
  nuisance,
  recalculation,
  allocation = c("exact", "approximate"),
  ...
)
```

Arguments

- `design`: Object of class `FarringtonManning` created by `setupFarringtonManning`.
- `n1`: Either the sample size of the first stage (if `recalculation = TRUE`) or the total sample size (if `recalculation = FALSE`).
- `nuisance`: Value of the nuisance parameter in (0,1). For the Farrington-Manning test this is the overall response rate.
- `recalculation`: Should the sample size be recalculated after `n1` patients are recruited?
- `allocation`: Whether the allocation ratio should be preserved exactly (`exact`) or approximately (`approximate`).
- `...`: Further optional arguments.

Details

The method is only vectorized in either `nuisance` or `n1`.

Value

One type I error rate value for every nuisance parameter and every value of `n1`.

Examples

```r
d <- setupFarringtonManning(alpha = 0.025, beta = 0.2, r = 1, delta = 0, delta_NI = 0.2)
toer(d, n1 = 20, nuisance = 0.25, recalculation = TRUE, allocation = "approximate")
```
**toer,Student-method**  

*Type I Error Rate*

**Description**

Computes the type I error rate of designs with blinded sample size recalculation or of fixed designs for one or several values of the nuisance parameter.

**Usage**

```r
## S4 method for signature 'Student'
toer(
  design,
  n1,
  nuisance,
  recalculation = TRUE,
  iters = 10000,
  seed = NULL,
  allocation = c("approximate", "exact"),
  ...
)
```

**Arguments**

- **design**: Object of class Student created by `setupStudent`
- **n1**: Either the sample size of the first stage (if `recalculation = TRUE`) or the total sample size (if `recalculation = FALSE`).
- **nuisance**: Value of the nuisance parameter. For the Student’s t-test this is the variance.
- **recalculation**: Should the sample size be recalculated after `n1` patients are recruited?
- **iters**: Number of simulation iterations.
- **seed**: Random seed for simulation.
- **allocation**: Whether the allocation ratio should be preserved exactly (exact) or approximately (approximate).
- **...**: Further optional arguments.

**Details**

The method is only vectorized in either `nuisance` or `n1`.

**Value**

One type I error rate value for every nuisance parameter and every value of `n1`. 
Examples

d <- setupStudent(alpha = .025, beta = .2, r = 1, delta = 3.5, delta_NI = 0, alternative = "greater", n_max = 156)
teor(d, n1 = 20, nuisance = 5.5, recalculation = TRUE)
Index

adjusted_alpha, 2, 8, 9, 24
adjusted_alpha, ChiSquare-method, 3
adjusted_alpha, FarringtonManning-method, 4
adjusted_alpha, Student-method, 6
blindrecalc, 7
ChiSquare, 3, 7, 8, 10, 15, 18, 25
ChiSquare (ChiSquare-class), 7
ChiSquare-class, 7

FarringtonManning, 3, 9, 10, 15, 18, 25
FarringtonManning
   (FarringtonManning-class), 9
FarringtonManning-class, 9

n_dist, 8, 9, 10, 24
n_dist, ChiSquare-method, 11
n_dist, FarringtonManning-method, 12
n_dist, Student-method, 13
n_fix, 8, 9, 14, 24
n_fix, ChiSquare-method, 15
n_fix, FarringtonManning-method, 16
n_fix, Student-method, 17

pow, 8, 9, 18, 24
pow, ChiSquare-method, 19
pow, FarringtonManning-method, 20
pow, Student-method, 21

setupChiSquare (ChiSquare-class), 7
setupFarringtonManning, 8, 9
setupFarringtonManning
   (FarringtonManning-class), 9
setupStudent, 22
setupStudent (Student-class), 23
simulation, 22
Student, 3, 10, 15, 18, 23–25
Student (Student-class), 23
Student-class, 23

toer, 8, 9, 24, 25
toer, ChiSquare-method, 26
toer, FarringtonManning-method, 27
toer, Student-method, 28