

# Package ‘blindrecalc’

June 17, 2025

**Type** Package

**Title** Blinded Sample Size Recalculation

**Version** 1.1.0

**Description** Computation of key characteristics and plots for blinded sample size recalculation. Continuous as well as binary endpoints are supported in superiority and non-inferiority trials. See Baumann, Pilz, Kieser (2022) <[doi:10.32614/RJ-2022-001](https://doi.org/10.32614/RJ-2022-001)> for a detailed description. The implemented methods include the approaches by Lu, K. (2019) <[doi:10.1002/pst.1737](https://doi.org/10.1002/pst.1737)>, Kieser, M. and Friede, T. (2000) <[doi:10.1002/\(SICI\)1097-0258\(20000415\)19:7%3C901::AID-SIM405%3E3.0.CO;2-L](https://doi.org/10.1002/(SICI)1097-0258(20000415)19:7%3C901::AID-SIM405%3E3.0.CO;2-L)>, Friede, T. and Kieser, M. (2004) <[doi:10.1002/pst.140](https://doi.org/10.1002/pst.140)>, Friede, T., Mitchell, C., Mueller-Veltorn, G. (2007) <[doi:10.1002/bimj.200610373](https://doi.org/10.1002/bimj.200610373)>, and Friede, T. and Kieser, M. (2011) <[doi:10.3414/ME09-01-0063](https://doi.org/10.3414/ME09-01-0063)>.

**License** MIT + file LICENSE

**URL** <https://github.com/imbi-heidelberg/blindrecalc>

**Encoding** UTF-8

**Suggests** testthat, covr, knitr, rmarkdown, bookdown

**Imports** methods, Rcpp

**Collate** test\_statistics.R methods.R Student.R blindrecalc.R  
ChiSquare.R ChiSquare\_helper.R FarringtonManning.R  
FarringtonManning\_helper.R RcppExports.R

**RoxygenNote** 7.3.2

**LinkingTo** Rcpp

**VignetteBuilder** knitr

**NeedsCompilation** yes

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

**Maintainer** Lukas Baumann <[baumann@imbi.uni-heidelberg.de](mailto:baumann@imbi.uni-heidelberg.de)>

**Repository** CRAN

**Date/Publication** 2025-06-17 08:00:02 UTC

Contents

adjusted_alpha . . . . .	2
adjusted_alpha,ChiSquare-method . . . . .	3
adjusted_alpha,FarringtonManning-method . . . . .	4
adjusted_alpha,Student-method . . . . .	6
blindrecalc . . . . .	7
ChiSquare-class . . . . .	8
FarringtonManning-class . . . . .	9
n_dist . . . . .	11
n_dist,ChiSquare-method . . . . .	12
n_dist,FarringtonManning-method . . . . .	13
n_dist,Student-method . . . . .	14
n_fix . . . . .	15
n_fix,ChiSquare-method . . . . .	16
n_fix,FarringtonManning-method . . . . .	17
n_fix,Student-method . . . . .	18
pow . . . . .	18
pow,ChiSquare-method . . . . .	19
pow,FarringtonManning-method . . . . .	20
pow,Student-method . . . . .	21
simulation . . . . .	22
Student-class . . . . .	23
toer . . . . .	25
toer,ChiSquare-method . . . . .	26
toer,FarringtonManning-method . . . . .	27
toer,Student-method . . . . .	28
<b>Index</b>	<b>30</b>

---

adjusted_alpha	<i>Adjusted level of significance</i>
----------------	---------------------------------------

---

Description

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

Usage

```
adjusted_alpha(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](#). 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, recalculation = TRUE,
               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,
  recalculation,
  nuis_ass,
  precision = 0.001,
  gamma = 0,
  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.
recalculation	Should the sample size be recalculated after n1 patients are recruited?
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.
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`  
*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 'FarringtonManning'
adjusted_alpha(
  design,
  n1,
  nuisance,
  recalculation,
  nuis_ass,
  precision = 0.001,
  gamma = 0,
  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?
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.
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**

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

```
## S4 method for signature 'Student'
adjusted_alpha(
  design,
  n1,
  nuisance,
  recalculation,
  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.
recalculation	Should the sample size be recalculated after n1 patients are recruited?
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

```
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, recalculation = TRUE,
              tol = 1e-4, iters = 1e3)
```

---

blindrecalc

*Blinded Sample Size Recalculation*

---

## Description

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

## Author(s)

**Maintainer:** Lukas Baumann <baumann@imbi.uni-heidelberg.de> ([ORCID](#))

Authors:

- Maximilian Pilz <maximilian.pilz@itwm.fraunhofer.de> ([ORCID](#))

Other contributors:

- Institute of Medical Biometry - University of Heidelberg [copyright holder]

## See Also

Useful links:

- <https://github.com/imbi-heidelberg/blindrecalc>

---

ChiSquare-class	<i>Chi-squared test</i>
-----------------	-------------------------

---

### 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

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

### Arguments

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

### Details

The nuisance parameter is the overall response probability  $p_0$ . In the blinded sample size #' recal-culation 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:

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

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.

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

### Value

An object of class `ChiSquare`.

### References

Friede, T., & Kieser, M. (2004). Sample size recalculation for binary data in internal pilot study designs. *Pharmaceutical Statistics: The Journal of Applied Statistics in the Pharmaceutical Industry*, 3(4), 269-279.

Kieser, M. (2020). *Methods and applications of sample size calculation and recalculation in clinical trials*. Springer.

### Examples

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

---

FarringtonManning-class

*Farrington Manning test*

---

### 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

```
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:

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

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.

**Value**

An object of class `FarringtonManning`.

**References**

Friede, T., Mitchell, C., & Mueller-Velten, G. (2007). Blinded sample size reestimation in non-inferiority trials with binary endpoints. *Biometrical Journal*, 49(6), 903-916.  
 Kieser, M. (2020). *Methods and applications of sample size calculation and recalculation in clinical trials*. Springer.

**Examples**

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

---

n_dist	<i>Distribution of the Sample Size</i>
--------	--

---

**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 <code>TestStatistic</code> created by <code>setup</code>
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 are 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**

```
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)
```

---

n\_dist,FarringtonManning-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 'FarringtonManning'
n_dist(
  design,
  n1,
  nuisance,
  summary,
  plot,
  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.
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).
...	Further optional arguments.

**Details**

Only sample sizes that occur with a probability of at least 0.01 are considered.

The method is only vectorized in either nuisance or n1.

**Value**

Summary and/or plot of the sample size distribution for each 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_dist(d, n1 = 30, nuisance = 0.2, summary = TRUE, plot = FALSE)
```

---

*n\_dist, Student-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 'Student'
n_dist(
  design,
  n1,
  nuisance,
  summary = TRUE,
  plot = FALSE,
  iters = 10000,
  seed = NULL,
  range = 0,
  allocation = c("approximate", "exact"),
  ...
)
```

**Arguments**

<code>design</code>	Object of class Student created by setupStudent.
<code>n1</code>	Either the sample size of the first stage (if recalculation = TRUE or the total sample size (if recalculation = FALSE).
<code>nuisance</code>	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	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

```
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

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

### 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,ChiSquare-method
<i>Fixed Sample Size</i>

---

**Description**

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

**Usage**

```
## S4 method for signature 'ChiSquare'
n_fix(
  design,
  nuisance,
  variance = c("heterogeneous", "homogeneous"),
  rounded = TRUE,
  ...
)
```

**Arguments**

design	Object of class ChiSquare created by setupChiSquare.
nuisance	Value of the nuisance parameter. For the Chi-Squared test this is the overall response rate.
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**

```
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**

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

---

n_fix, Student-method	<i>Fixed Sample Size</i>
-----------------------	--------------------------

---

**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)
```

---

pow	<i>Power</i>
-----	--------------

---

**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**

```
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**

```
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)
```

---

pow,ChiSquare-method	<i>Power</i>
----------------------	--------------

---

**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**

```
## 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**

```
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)
```

---

pow,FarringtonManning-method  
*Power*

---

**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**

```
## 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**

```
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)
```

---

pow, Student-method	<i>Power</i>
---------------------	--------------

---

**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**

```
## 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**

```
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

The implementation follows the algorithm in Lu (2019): Distribution of the two-sample t-test statistic following blinded sample size re-estimation. *Pharmaceutical Statistics* 15: 208-215. Since Lu (2019) assumes negative non-inferiority margins, the non-inferiority margin of design is multiplied with -1 internally.

### 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

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

### Arguments

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

Lu, K. (2019). Distribution of the two-sample t-test statistic following blinded sample size re-estimation. *Pharmaceutical Statistics* 15(3): 208-215.

### Examples

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

---

toer	<i>Type I Error Rate</i>
------	--------------------------

---

### 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 <code>TestStatistic</code> created by <code>setup</code>
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 = .025, beta = .2, r = 1, delta = 3.5, delta_NI = 0,
                  alternative = "greater", n_max = 156)
toer(d, n1 = 20, nuisance = 5.5, recalculation = TRUE)
```

---

toer, ChiSquare-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**

```
## 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**

```
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**

```
## 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**

```
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	<i>Type I Error Rate</i>
----------------------	--------------------------

---

**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**

```
## 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)  
toer(d, n1 = 20, nuisance = 5.5, recalculation = TRUE)
```

# Index

adjusted\_alpha, [2](#), [9](#), [25](#)  
adjusted\_alpha, ChiSquare-method, [3](#)  
adjusted\_alpha, FarringtonManning-method, [4](#)  
adjusted\_alpha, Student-method, [6](#)  
  
blindrecalc, [7](#)  
blindrecalc-package (blindrecalc), [7](#)  
  
ChiSquare, [3](#), [8](#), [9](#), [11](#), [16](#), [19](#), [25](#)  
ChiSquare (ChiSquare-class), [8](#)  
ChiSquare-class, [8](#)  
  
FarringtonManning, [3](#), [9–11](#), [16](#), [19](#), [25](#)  
FarringtonManning  
    (FarringtonManning-class), [9](#)  
FarringtonManning-class, [9](#)  
  
n\_dist, [9](#), [11](#), [25](#)  
n\_dist, ChiSquare-method, [12](#)  
n\_dist, FarringtonManning-method, [13](#)  
n\_dist, Student-method, [14](#)  
n\_fix, [9](#), [15](#), [25](#)  
n\_fix, ChiSquare-method, [16](#)  
n\_fix, FarringtonManning-method, [17](#)  
n\_fix, Student-method, [18](#)  
  
pow, [9](#), [18](#), [25](#)  
pow, ChiSquare-method, [19](#)  
pow, FarringtonManning-method, [20](#)  
pow, Student-method, [21](#)  
  
setupChiSquare (ChiSquare-class), [8](#)  
setupFarringtonManning, [9](#)  
setupFarringtonManning  
    (FarringtonManning-class), [9](#)  
setupStudent, [22](#)  
setupStudent (Student-class), [23](#)  
simulation, [22](#)  
Student, [3](#), [11](#), [16](#), [19](#), [24](#), [25](#)  
Student (Student-class), [23](#)  
  
Student-class, [23](#)  
  
toer, [9](#), [25](#), [25](#)  
toer, ChiSquare-method, [26](#)  
toer, FarringtonManning-method, [27](#)  
toer, Student-method, [28](#)