

Package ‘bbssr’

August 20, 2026

Type Package

Title Blinded Sample Size Re-Estimation for Binary Endpoints

Version 2.0.0

Description Tools for blinded sample size re-estimation (BSSR) in two-arm clinical trials with binary endpoints, together with the exact power and sample size calculations that the re-estimation relies on. Five exact statistical tests are implemented: Pearson chi-squared, Fisher exact, Fisher mid-p, Z-pooled exact unconditional, and Boschloo exact unconditional. Each test is available with a one-sided or a two-sided alternative, and the exact unconditional tests can be combined with the Berger-Boos procedure. Sample sizes can be re-estimated either at the planning stage, to study the operating characteristics of a design, or from the blinded data of a trial that is under way. Statistical methods based on Mehrotra et al. (2003) <[doi:10.1111/1541-0420.00051](https://doi.org/10.1111/1541-0420.00051)>, Berger and Boos (1994) <[doi:10.1080/01621459.1994.10476836](https://doi.org/10.1080/01621459.1994.10476836)> and Kieser (2020) <[doi:10.1007/978-3-030-49528-2_21](https://doi.org/10.1007/978-3-030-49528-2_21)>.

License MIT + file LICENSE

URL <https://github.com/gosukehommaEX/bbssr>

BugReports <https://github.com/gosukehommaEX/bbssr/issues>

Depends R (>= 3.5.0)

Imports fpCompare, ggplot2 (>= 3.4.0), Rcpp, stats, utils

LinkingTo Rcpp

Suggests testthat (>= 3.0.0), knitr, rmarkdown, Exact, exact2x2, microbenchmark

VignetteBuilder knitr

Encoding UTF-8

Language en-GB

Config/testthat/edition 3

Config/roxygen2/version 8.0.0

NeedsCompilation yes

Author Gosuke Homma [aut, cre]

Maintainer Gosuke Homma <my.name.is.gosuke@gmail.com>
Repository CRAN
Date/Publication 2026-08-20 11:02:11 UTC

Contents

BinaryBSSR	2
BinaryPower	5
BinaryPowerBSSR	6
BinaryRR	9
BinarySampleSize	11
plot.bbssr_power	13
plot.bbssr_powerbssr	14
plot.bbssr_rr	16
plot.bbssr_samplesize	17
print.bbssr_bssr	19
print.bbssr_power	19
print.bbssr_powerbssr	20
print.bbssr_rr	21
print.bbssr_samplesize	21
Index	23

BinaryBSSR	<i>Sample Size Re-estimation from Observed Blinded Interim Data</i>
------------	---

Description

Re-estimates the sample size of an ongoing two-arm trial with a binary endpoint from the blinded data available at an interim analysis, and reports how many patients still have to be enrolled in each group during the second stage. Only the total number of patients and the total number of responders are required, so the treatment allocation remains concealed.

Usage

```
BinaryBSSR(  
  n1,  
  n2,  
  S,  
  Delta.A,  
  r,  
  alpha,  
  tar.power,  
  Test,  
  restricted = FALSE,  
  N1 = NULL,
```

```

N2 = NULL,
alternative = c("greater", "two.sided"),
tsmethod = c("minlike", "central"),
n.grid = 100,
bb.gamma = 0
)

```

Arguments

n1	Number of patients of group 1 observed at the interim analysis
n2	Number of patients of group 2 observed at the interim analysis
S	Total number of responders observed at the interim analysis, pooled over both groups
Delta.A	Assumed treatment effect (risk difference) used to split the blinded pooled proportion into group-specific proportions
r	Allocation ratio to group 1 (i.e., allocation ratio of group 1:group 2 = r:1, r > 0)
alpha	Level of significance for the alternative specified by alternative
tar.power	Target power
Test	Type of statistical test. Options: 'Chisq', 'Fisher', 'Fisher-midP', 'Z-pool', or 'Boschloo'
restricted	Logical. If TRUE, the re-estimated sample size is not allowed to fall below the planned sample size given by N1 and N2. Default is FALSE
N1	Planned sample size of group 1. Required when restricted is TRUE
N2	Planned sample size of group 2. Required when restricted is TRUE
alternative	Direction of the alternative hypothesis. Options: 'greater' (default) or 'two.sided'
tsmethod	Convention used to construct the two-sided version of the conditional tests. Options: 'minlike' (default) or 'central'
n.grid	Number of grid points used to search over the nuisance parameter of the unconditional tests. Default is 100
bb.gamma	Confidence level parameter of the Berger-Boos procedure. The default of 0 disables the procedure

Details

The blinded estimate of the pooled response probability is $\hat{p} = S / (n1 + n2)$. Group-specific proportions are recovered as $\hat{p}_1 = \hat{p} + \Delta.A / (1 + r)$ and $\hat{p}_2 = \hat{p} - r \Delta.A / (1 + r)$, truncated to the unit interval, and the sample size is then re-estimated by [BinarySampleSize](#).

Under the unrestricted rule the final size of group 2 is the larger of the re-estimated size and what has already been observed. Under the restricted rule it is raised to the planned size first, so the trial can only grow. The final size of group 1 is then $\text{ceiling}(r N2.\text{final})$, so an imbalance already present at the interim is corrected by the remaining enrolment instead of being carried forward. Neither second-stage size is ever negative.

While [BinaryPowerBSSR](#) evaluates the operating characteristics of a BSSR design at the planning stage, this function is applied once, to the data of a trial that is under way.

Value

An object of class `bbssr_bssr`, a data frame with one row containing:

n1 Interim sample size of group 1
n2 Interim sample size of group 2
n Total interim sample size
S Total number of interim responders
hat.p Blinded estimate of the pooled response probability
hat.p1 Recovered response probability of group 1
hat.p2 Recovered response probability of group 2
N1.re Re-estimated total sample size of group 1
N2.re Re-estimated total sample size of group 2
N.re Re-estimated total sample size
n1.stage2 Number of additional patients to enrol in group 1
n2.stage2 Number of additional patients to enrol in group 2
n.stage2 Total number of additional patients to enrol
N1.final Final sample size of group 1
N2.final Final sample size of group 2
N.final Final total sample size
Power Exact power at the final sample size under the recovered proportions

Author(s)

Gosuke Homma (<my.name.is.gosuke@gmail.com>)

See Also

[BinaryPowerBSSR](#), [BinarySampleSize](#)

Examples

```
# Interim data: 20 patients per group, 11 responders in total
BinaryBSSR(n1 = 20, n2 = 20, S = 11, Delta.A = 0.3, r = 1,
            alpha = 0.025, tar.power = 0.8, Test = 'Chisq')

# Restricted rule with a planned sample size of 40 per group
BinaryBSSR(n1 = 20, n2 = 20, S = 11, Delta.A = 0.3, r = 1,
            alpha = 0.025, tar.power = 0.8, Test = 'Boschloo',
            restricted = TRUE, N1 = 40, N2 = 40)
```

Description

Calculates power for two-arm trials with binary endpoints using exact statistical tests. Five tests are supported, each of which can be applied with a one-sided or a two-sided alternative, and vectors of response probabilities are accepted.

Usage

```
BinaryPower(
  p1,
  p2,
  N1,
  N2,
  alpha,
  Test,
  alternative = c("greater", "two.sided"),
  tsmethod = c("minlike", "central"),
  n.grid = 100,
  bb.gamma = 0
)
```

Arguments

p1	True probability of responders for group 1 (can be a vector)
p2	True probability of responders for group 2 (can be a vector of the same length as p1)
N1	Sample size for group 1
N2	Sample size for group 2
alpha	Level of significance for the alternative specified by alternative
Test	Type of statistical test. Options: 'Chisq', 'Fisher', 'Fisher-midP', 'Z-pool', or 'Boschloo'
alternative	Direction of the alternative hypothesis. Options: 'greater' (default) or 'two.sided'
tsmethod	Convention used to construct the two-sided version of the conditional tests. Options: 'minlike' (default) or 'central'
n.grid	Number of grid points used to search over the nuisance parameter of the unconditional tests. Default is 100
bb.gamma	Confidence level parameter of the Berger-Boos procedure. The default of 0 disables the procedure

Details

The power is obtained by summing the joint probability mass function of the two independent binomial counts over the rejection region returned by [BinaryRR](#). The summation covers the whole rejection region rather than a row-wise tail, so it remains valid for two-sided tests, whose rejection regions are not contiguous within a row of the outcome grid.

Value

An object of class `bbssr_power`, a data frame with one row per element of `p1` containing:

p1 True probability of responders for group 1
p2 True probability of responders for group 2
N1 Sample size for group 1
N2 Sample size for group 2
alpha Level of significance
Test Name of the statistical test
alternative Direction of the alternative hypothesis
Power Exact power

Author(s)

Gosuke Homma (<my.name.is.gosuke@gmail.com>)

Examples

```
# Power of the one-sided chi-squared test
BinaryPower(p1 = 0.5, p2 = 0.2, N1 = 5, N2 = 5, alpha = 0.025, Test = 'Chisq')

# Power over a range of response probabilities for the two-sided Boschloo test
pw <- BinaryPower(p1 = c(0.5, 0.6, 0.7, 0.8), p2 = rep(0.2, 4),
                  N1 = 20, N2 = 20, alpha = 0.05, Test = 'Boschloo',
                  alternative = 'two.sided')

print(pw)
plot(pw)
```

BinaryPowerBSSR

Power of a Blinded Sample Size Re-estimation Design for Binary End-points

Description

Calculates the power of a two-arm trial with a binary endpoint when blinded sample size re-estimation (BSSR) is implemented, together with the power of the corresponding fixed-sample design. Five exact tests are supported, each of which can be applied with a one-sided or a two-sided alternative, under either a restricted or an unrestricted design rule.

Usage

```

BinaryPowerBSSR(
  p,
  Delta.A,
  Delta.T,
  N1,
  N2,
  omega,
  r,
  alpha,
  tar.power,
  Test,
  restricted = FALSE,
  alternative = c("greater", "two.sided"),
  tsmethod = c("minlike", "central"),
  n.grid = 100,
  bb.gamma = 0
)

```

Arguments

<code>p</code>	Vector of true pooled proportions of responders from both groups
<code>Delta.A</code>	Assumed treatment effect (risk difference)
<code>Delta.T</code>	True treatment effect (risk difference)
<code>N1</code>	Initial sample size of group 1
<code>N2</code>	Initial sample size of group 2
<code>omega</code>	Fraction of the initial sample size observed at the interim analysis. The interim size of group 2 is <code>ceiling(omega N2)</code> and that of group 1 is <code>ceiling(r ceiling(omega N2))</code> , so the interim analysis keeps the allocation ratio
<code>r</code>	Allocation ratio to group 1
<code>alpha</code>	Level of significance for the alternative specified by <code>alternative</code>
<code>tar.power</code>	Target power
<code>Test</code>	Type of statistical test. Options: 'Chisq', 'Fisher', 'Fisher-midP', 'Z-pool', or 'Boschloo'
<code>restricted</code>	Logical. If TRUE, the re-estimated sample size is not allowed to fall below the initial sample size. Default is FALSE
<code>alternative</code>	Direction of the alternative hypothesis. Options: 'greater' (default) or 'two.sided'
<code>tsmethod</code>	Convention used to construct the two-sided version of the conditional tests. Options: 'minlike' (default) or 'central'
<code>n.grid</code>	Number of grid points used to search over the nuisance parameter of the unconditional tests. Default is 100
<code>bb.gamma</code>	Confidence level parameter of the Berger-Boos procedure. The default of 0 disables the procedure

Details

Both the interim and the final sample size of group 1 are obtained from the size of group 2 by a single application of `ceiling(r ...)`, so the allocation ratio is preserved as closely as whole numbers allow and is exact whenever r is a whole number. The size of the second stage follows as the difference between the final and the interim size, which keeps the two stages adding up to the final size. The argument `N1` enters only through the fixed-sample comparator, and a warning is issued when it is not `ceiling(r N2)`.

At the interim analysis the pooled number of responders is observed without unblinding. The pooled proportion is combined with the assumed treatment effect `Delta.A` to recover group-specific proportions, from which the sample size is re-estimated. The power is then averaged over the distribution of the interim outcome.

Setting `Delta.T` to 0 makes the two groups identical, so `power.BSSR` and `power.TRAD` become rejection probabilities under the null hypothesis. They are evaluated only at the values supplied through `p`, since the function does not search over the unit interval on its own. The largest type I error rate of the design is therefore obtained by passing a grid such as `p = seq(0.02, 0.98, by = 0.02)` and taking the maximum of the `power.BSSR` column. The rejection region and the re-estimated sample size are shared across the elements of `p`, so a fine grid costs much less than the same number of separate calls.

The weighted approach available in earlier versions of the package has been removed.

Value

An object of class `bbssr_powerbssr`, a data frame with one row per element of `p` containing:

- p1** True probability of responders for group 1
- p2** True probability of responders for group 2
- p** True pooled probability of responders from both groups
- power.BSSR** Power of the BSSR design
- power.TRAD** Power of the fixed-sample design
- E.N** Expected total sample size of the BSSR design

The interim sample sizes are stored as the attributes `n1.interim` and `n2.interim`.

Author(s)

Gosuke Homma (<my.name.is.gosuke@gmail.com>)

Examples

```
# Small BSSR calculation with the chi-squared test
BinaryPowerBSSR(
  p = 0.45,
  Delta.A = 0.3, Delta.T = 0.3,
  N1 = 5, N2 = 5, omega = 0.5, r = 1,
  alpha = 0.025, tar.power = 0.8, Test = 'Chisq'
)
```

```
res <- BinaryPowerBSSR(
  p = seq(0.19, 0.37, by = 0.03),
  Delta.A = 0.36, Delta.T = 0.36,
  N1 = 24, N2 = 24, omega = 0.5, r = 1,
  alpha = 0.025, tar.power = 0.8, Test = 'Z-pool'
)
print(res)
plot(res)
```

BinaryRR

Rejection Region for Two-Arm Trials with Binary Endpoints

Description

Provides a rejection region (RR) for two-arm trials with binary endpoints using various exact statistical tests. Five tests are supported, each of which can be applied with a one-sided or a two-sided alternative.

Usage

```
BinaryRR(
  N1,
  N2,
  alpha,
  Test,
  alternative = c("greater", "two.sided"),
  tsmethod = c("minlike", "central"),
  n.grid = 100,
  bb.gamma = 0
)
```

Arguments

N1	Sample size for group 1
N2	Sample size for group 2
alpha	Level of significance for the alternative specified by alternative
Test	Type of statistical test. Options: 'Chisq', 'Fisher', 'Fisher-midP', 'Z-pool', or 'Boschloo'
alternative	Direction of the alternative hypothesis. Options: 'greater' (default) for the one-sided alternative that the response probability of group 1 exceeds that of group 2, or 'two.sided'
tsmethod	Convention used to construct the two-sided version of the conditional tests. Options: 'minlike' (default) or 'central'. Ignored when alternative is 'greater', and ignored by 'Chisq' and 'Z-pool', whose two-sided versions are based on the absolute value of the Z statistic

<code>n.grid</code>	Number of grid points used to search over the nuisance parameter of the unconditional tests. Default is 100. Ignored by the conditional tests
<code>bb.gamma</code>	Confidence level parameter of the Berger-Boos procedure for the unconditional tests. The default of 0 disables the procedure. A positive value restricts the search over the nuisance parameter to an exact 100 (1 - <code>bb.gamma</code>) percent confidence interval and adds <code>bb.gamma</code> to the resulting p-value. A common choice is 0.0001

Details

The function supports the following five tests:

- The Pearson chi-squared test (`Chisq`)
- The Fisher exact test (`Fisher`)
- The Fisher mid-p test (`Fisher-midP`)
- The Z-pooled exact unconditional test (`Z-pool`)
- The Boschloo exact unconditional test (`Boschloo`)

For the two-sided versions of the conditional tests, `'minlike'` sums the null probabilities of all tables that are no more likely than the observed table, which is the convention of `stats::fisher.test`, whereas `'central'` doubles the smaller of the two one-sided tail probabilities. The two-sided versions of `'Chisq'` and `'Z-pool'` order the outcomes by the absolute value of the Z statistic.

The unconditional tests maximize the null tail probability of an ordering statistic over the common response probability, which is a nuisance parameter. Outcomes sharing the same value of the ordering statistic receive the same p-value.

Value

An object of class `bbssr_rr`, which is a logical matrix of dimension $(N1 + 1)$ by $(N2 + 1)$ whose entry `[i + 1, j + 1]` is TRUE when the null hypothesis is rejected at `i` responders in group 1 and `j` responders in group 2. The design settings are stored as attributes

Author(s)

Gosuke Homma (<my.name.is.gosuke@gmail.com>)

Examples

```
# Simple example with small sample sizes
RR <- BinaryRR(N1 = 5, N2 = 5, alpha = 0.025, Test = 'Chisq')
print(RR)

# Two-sided Boschloo test with the Berger-Boos procedure
RR <- BinaryRR(N1 = 20, N2 = 10, alpha = 0.05, Test = 'Boschloo',
               alternative = 'two.sided', bb.gamma = 0.0001)
print(RR)
plot(RR)
```

Description

Calculates the required sample size for two-arm trials with binary endpoints using exact statistical tests. Five tests are supported, each of which can be applied with a one-sided or a two-sided alternative.

Usage

```
BinarySampleSize(
  p1,
  p2,
  r,
  alpha,
  tar.power,
  Test,
  alternative = c("greater", "two.sided"),
  tsmethod = c("minlike", "central"),
  n.grid = 100,
  bb.gamma = 0
)
```

Arguments

p1	True probability of responders for group 1
p2	True probability of responders for group 2
r	Allocation ratio to group 1 (i.e., allocation ratio of group 1:group 2 = r:1, r > 0)
alpha	Level of significance for the alternative specified by alternative
tar.power	Target power
Test	Type of statistical test. Options: 'Chisq', 'Fisher', 'Fisher-midP', 'Z-pool', or 'Boschloo'
alternative	Direction of the alternative hypothesis. Options: 'greater' (default) or 'two.sided'
tsmethod	Convention used to construct the two-sided version of the conditional tests. Options: 'minlike' (default) or 'central'
n.grid	Number of grid points used to search over the nuisance parameter of the unconditional tests. Default is 100
bb.gamma	Confidence level parameter of the Berger-Boos procedure. The default of 0 disables the procedure

Details

The calculation uses a three-step approach:

1. Calculate an initial sample size from the normal approximation to the chi-squared test
2. Evaluate the exact power at the initial sample size
3. Move the sample size up or down one unit at a time until the smallest sample size attaining the target power is found

The normal approximation of the first step uses α for a one-sided alternative and $\alpha / 2$ for a two-sided alternative. Only the starting value of the search is affected, so the returned sample size is exact in either case.

Value

An object of class `bbsr_samplesize`, a data frame with one row containing:

p1 True probability of responders for group 1
p2 True probability of responders for group 2
r Allocation ratio to group 1
alpha Level of significance
tar.power Target power
Test Name of the statistical test
alternative Direction of the alternative hypothesis
Power Exact power at the selected sample size
N1 Required sample size of group 1
N2 Required sample size of group 2
N Total required sample size

Author(s)

Gosuke Homma (<my.name.is.gosuke@gmail.com>)

Examples

```
# One-sided chi-squared test
BinarySampleSize(p1 = 0.4, p2 = 0.2, r = 1, alpha = 0.025,
                 tar.power = 0.8, Test = 'Chisq')

# Two-sided Fisher exact test
BinarySampleSize(p1 = 0.5, p2 = 0.2, r = 2, alpha = 0.05,
                 tar.power = 0.9, Test = 'Fisher', alternative = 'two.sided')
```

plot.bbssr_power	<i>Plot Exact Power against the Response Probability of Group 1</i>
------------------	---

Description

Plot Exact Power against the Response Probability of Group 1

Usage

```
## S3 method for class 'bbssr_power'
plot(
  x,
  main = NULL,
  sub = NULL,
  xlab = NULL,
  ylab = NULL,
  ylim = c(0, 1),
  show.points = TRUE,
  colours = NULL,
  base_size = 11,
  ...
)
```

Arguments

x	An object of class <code>bbssr_power</code> returned by BinaryPower
main	Title of the plot. The default of <code>NULL</code> builds the title from the object and a single NA drops it
sub	Subtitle of the plot, following the convention of <code>main</code>
xlab	Label of the horizontal axis, following the convention of <code>main</code>
ylab	Label of the vertical axis, following the convention of <code>main</code>
ylim	Numeric vector of length two giving the range of the vertical axis, or <code>NULL</code> for a range chosen from the data
show.points	Logical. If <code>TRUE</code> , the default, a marker is drawn at every evaluated response probability. Set it to <code>FALSE</code> to draw the curve alone, which is easier to read on a fine grid
colours	Colour of the curve
base_size	Base font size of the theme, in points
...	Further arguments, currently ignored

Details

The vertical range is imposed with [coord_cartesian](#), so points outside `ylim` are hidden rather than removed from the data.

Value

A ggplot object

Examples

```
pw <- BinaryPower(p1 = seq(0.3, 0.8, by = 0.1), p2 = rep(0.2, 6),
                  N1 = 20, N2 = 20, alpha = 0.025, Test = 'Chisq')
plot(pw)

# Enlarge the type, rescale the vertical axis, rename the axes and drop the markers
plot(pw, base_size = 14, ylim = c(0.2, 1), sub = NA, show.points = FALSE,
     xlab = 'Response probability, experimental group', ylab = 'Exact power')
```

plot.bbssr_powerbssr *Plot the Power of a BSSR Design against the Fixed-Sample Design*

Description

Plot the Power of a BSSR Design against the Fixed-Sample Design

Usage

```
## S3 method for class 'bbssr_powerbssr'
plot(
  x,
  main = NULL,
  sub = NULL,
  xlab = NULL,
  ylab = NULL,
  ylim = NULL,
  ref.line = NULL,
  legend.title = NULL,
  legend.labels = NULL,
  show.points = TRUE,
  colours = NULL,
  base_size = 11,
  ...
)
```

Arguments

x	An object of class bbssr_powerbssr returned by BinaryPowerBSSR
main	Title of the plot. The default of NULL builds the title from the object and a single NA drops it
sub	Subtitle of the plot, following the convention of main
xlab	Label of the horizontal axis, following the convention of main

ylab	Label of the vertical axis, following the convention of main
ylim	Numeric vector of length two giving the range of the vertical axis, or NULL for a range chosen from the data
ref.line	Numeric vector of heights at which a dashed horizontal line is drawn. The default of NULL places one line at the target power and a single NA draws no line
legend.title	Title of the legend. The default of NULL leaves the legend untitled
legend.labels	Character vector of length two replacing the entries of the legend, in the order BSSR then fixed sample
show.points	Logical. If TRUE, the default, a marker is drawn at every evaluated pooled probability. Set it to FALSE to draw the curves alone, which is easier to read on a fine grid
colours	Character vector of length two giving the colour of the two curves, in the order BSSR then fixed sample
base_size	Base font size of the theme, in points
...	Further arguments, currently ignored

Details

Setting `Delta.T` to 0 in `BinaryPowerBSSR` turns the two power columns into rejection probabilities under the null hypothesis. The default reference line at the target power is then out of place, and the level of significance passed through `ref.line` together with a rescaled `ylim` gives a readable plot of the type I error rate.

The vertical range is imposed with `coord_cartesian`, so points outside `ylim` are hidden rather than removed from the data.

Value

A ggplot object

Examples

```
res <- BinaryPowerBSSR(
  p = seq(0.19, 0.37, by = 0.03),
  Delta.A = 0.36, Delta.T = 0.36,
  N1 = 24, N2 = 24, omega = 0.5, r = 1,
  alpha = 0.025, tar.power = 0.8, Test = 'Chisq'
)
plot(res)

# Type I error rate, with the reference line moved to the level of significance
tie <- BinaryPowerBSSR(
  p = seq(0.19, 0.37, by = 0.03),
  Delta.A = 0.36, Delta.T = 0,
  N1 = 24, N2 = 24, omega = 0.5, r = 1,
  alpha = 0.025, tar.power = 0.8, Test = 'Chisq'
)
plot(tie, ref.line = 0.025, ylim = c(0.015, 0.030), base_size = 14,
  show.points = FALSE,
```

```
main = 'Type I error rate of the BSSR design', ylab = 'Type I error rate',
legend.title = 'Design', legend.labels = c('BSSR', 'Fixed'))
```

plot.bbssr_rr

Plot a Rejection Region

Description

Displays the outcome grid of a two-arm trial with a binary endpoint and shades the outcomes for which the null hypothesis is rejected.

Usage

```
## S3 method for class 'bbssr_rr'
plot(
  x,
  main = NULL,
  sub = NULL,
  xlab = NULL,
  ylab = NULL,
  legend.title = NULL,
  legend.labels = NULL,
  colours = NULL,
  base_size = 11,
  ...
)
```

Arguments

x	An object of class <code>bbssr_rr</code> returned by BinaryRR
main	Title of the plot. The default of <code>NULL</code> builds the title from the object and a single NA drops it
sub	Subtitle of the plot, following the convention of <code>main</code>
xlab	Label of the horizontal axis, following the convention of <code>main</code>
ylab	Label of the vertical axis, following the convention of <code>main</code>
legend.title	Title of the legend. The default of <code>NULL</code> leaves the legend untitled
legend.labels	Character vector of length two replacing the entries of the legend, in the order retained then rejected
colours	Character vector of length two giving the fill of the tiles, in the order retained then rejected
base_size	Base font size of the theme, in points
...	Further arguments, currently ignored

Details

Both axes count responders, so the breaks are restricted to integers. The vertical axis runs downwards, which places the outcome with no responders in the top left corner and matches the layout of the map printed by `print.bbssr_rr`.

Value

A ggplot object

Examples

```
RR <- BinaryRR(N1 = 10, N2 = 10, alpha = 0.025, Test = 'Chisq')
plot(RR)

# Enlarge the type and relabel the legend
plot(RR, base_size = 14, legend.title = 'Decision',
      legend.labels = c('do not reject', 'reject'),
      colours = c('white', 'grey30'))
```

`plot.bbssr_samplesize` *Plot the Exact Power Curve around a Sample Size Solution*

Description

Recomputes the exact power over a range of sample sizes of group 2 and marks the selected sample size and the target power.

Usage

```
## S3 method for class 'bbssr_samplesize'
plot(
  x,
  N2.range = NULL,
  main = NULL,
  sub = NULL,
  xlab = NULL,
  ylab = NULL,
  ylim = NULL,
  ref.line = NULL,
  ref.line.N2 = NULL,
  show.points = TRUE,
  colours = NULL,
  base_size = 11,
  ...
)
```

Arguments

<code>x</code>	An object of class <code>bbssr_samplesize</code> returned by <code>BinarySampleSize</code>
<code>N2.range</code>	Optional integer vector of sample sizes of group 2 at which the power is evaluated. By default the selected sample size plus or minus five is used
<code>main</code>	Title of the plot. The default of <code>NULL</code> builds the title from the object and a single <code>NA</code> drops it
<code>sub</code>	Subtitle of the plot, following the convention of <code>main</code>
<code>xlab</code>	Label of the horizontal axis, following the convention of <code>main</code>
<code>ylab</code>	Label of the vertical axis, following the convention of <code>main</code>
<code>ylim</code>	Numeric vector of length two giving the range of the vertical axis, or <code>NULL</code> for a range chosen from the data
<code>ref.line</code>	Numeric vector of heights at which a dashed horizontal line is drawn. The default of <code>NULL</code> places one line at the target power and a single <code>NA</code> draws no line
<code>ref.line.N2</code>	Numeric vector of positions at which a dotted vertical line is drawn. The default of <code>NULL</code> places one line at the selected sample size and a single <code>NA</code> draws no line
<code>show.points</code>	Logical. If <code>TRUE</code> , the default, a marker is drawn at every evaluated sample size. Set it to <code>FALSE</code> to draw the curve alone, which is easier to read over a wide range
<code>colours</code>	Colour of the curve
<code>base_size</code>	Base font size of the theme, in points
<code>...</code>	Further arguments, currently ignored

Details

The power is recomputed at every point of `N2.range`, so a wide range combined with one of the unconditional tests can take a long time to evaluate.

The vertical range is imposed with `coord_cartesian`, so points outside `ylim` are hidden rather than removed from the data.

Value

A `ggplot` object

Examples

```
ss <- BinarySampleSize(p1 = 0.4, p2 = 0.2, r = 1, alpha = 0.025,
                      tar.power = 0.8, Test = 'Chisq')
plot(ss)

# Enlarge the type, rename the axes and draw neither reference line
plot(ss, base_size = 14, ref.line = NA, ref.line.N2 = NA,
     xlab = 'Sample size of the control group', ylab = 'Exact power')
```

```
print.bbssr_bbssr      Print a Sample Size Re-estimation from Interim Data
```

Description

Print a Sample Size Re-estimation from Interim Data

Usage

```
## S3 method for class 'bbssr_bbssr'
print(x, digits = 4, ...)
```

Arguments

x	An object of class bbssr_bbssr returned by BinaryBSSR
digits	Number of significant digits used for the proportions and the power
...	Further arguments, currently ignored

Value

The object x, returned invisibly

Examples

```
res <- BinaryBSSR(n1 = 20, n2 = 20, S = 11, Delta.A = 0.3, r = 1,
                  alpha = 0.025, tar.power = 0.8, Test = 'Chisq')
print(res)
```

```
print.bbssr_power      Print Exact Power Results
```

Description

Print Exact Power Results

Usage

```
## S3 method for class 'bbssr_power'
print(x, digits = 4, ...)
```

Arguments

x	An object of class bbssr_power returned by BinaryPower
digits	Number of significant digits used for the power
...	Further arguments, currently ignored

Value

The object x, returned invisibly

Examples

```
pw <- BinaryPower(p1 = 0.5, p2 = 0.2, N1 = 5, N2 = 5, alpha = 0.025, Test = 'Chisq')
print(pw)
```

```
print.bbssr_powerbssr Print the Operating Characteristics of a BSSR Design
```

Description

Print the Operating Characteristics of a BSSR Design

Usage

```
## S3 method for class 'bbssr_powerbssr'
print(x, digits = 4, ...)
```

Arguments

x	An object of class <code>bbssr_powerbssr</code> returned by BinaryPowerBSSR
digits	Number of significant digits used for the power
...	Further arguments, currently ignored

Value

The object x, returned invisibly

Examples

```
res <- BinaryPowerBSSR(
  p = 0.45,
  Delta.A = 0.3, Delta.T = 0.3,
  N1 = 5, N2 = 5, omega = 0.5, r = 1,
  alpha = 0.025, tar.power = 0.8, Test = 'Chisq'
)
print(res)
```

```
print.bbssr_rr
```

Print a Rejection Region

Description

Print a Rejection Region

Usage

```
## S3 method for class 'bbssr_rr'
print(x, show.map = NULL, ...)
```

Arguments

x	An object of class bbssr_rr returned by BinaryRR
show.map	Logical. If TRUE, the outcome grid is printed as a map in which X marks rejection of the null hypothesis. The default prints the map only when the grid has at most 400 cells
...	Further arguments, currently ignored

Value

The object x, returned invisibly

Examples

```
RR <- BinaryRR(N1 = 5, N2 = 5, alpha = 0.025, Test = 'Chisq')
print(RR)
```

```
print.bbssr_samplesize
```

Print a Sample Size Calculation

Description

Print a Sample Size Calculation

Usage

```
## S3 method for class 'bbssr_samplesize'
print(x, digits = 4, ...)
```

Arguments

<code>x</code>	An object of class <code>bbssr_samplesize</code> returned by BinarySampleSize
<code>digits</code>	Number of significant digits used for the power
<code>...</code>	Further arguments, currently ignored

Value

The object `x`, returned invisibly

Examples

```
ss <- BinarySampleSize(p1 = 0.4, p2 = 0.2, r = 1, alpha = 0.025,  
                      tar.power = 0.8, Test = 'Chisq')  
print(ss)
```

Index

BinaryBSSR, [2](#), [19](#)
BinaryPower, [5](#), [13](#), [19](#)
BinaryPowerBSSR, [3](#), [4](#), [6](#), [14](#), [15](#), [20](#)
BinaryRR, [6](#), [9](#), [16](#), [21](#)
BinarySampleSize, [3](#), [4](#), [11](#), [18](#), [22](#)

coord_cartesian, [13](#), [15](#), [18](#)

plot.bbssr_power, [13](#)
plot.bbssr_powerbssr, [14](#)
plot.bbssr_rr, [16](#)
plot.bbssr_samplesize, [17](#)
print.bbssr_bssr, [19](#)
print.bbssr_power, [19](#)
print.bbssr_powerbssr, [20](#)
print.bbssr_rr, [17](#), [21](#)
print.bbssr_samplesize, [21](#)