Package: bfbin2arm (via r-universe)

October 28, 2026

Type Package

Title Bayes Factor Design for Two-Arm Binomial Trials

Version 0.1.7

Date 2026-10-27

Description Design and analysis of one- and two-stage clinical trials using Bayes factors. Implements methods for binomial and continuous endpoints, including point-null and directional hypotheses, power calculations, and sample size calibration. Supports one-arm trials with a single treatment arm and two-arm trials with treatment and control arms. Includes methods for equivalence, non-inferiority, and superiority designs.

Depends R (>= 4.0.0)

Imports stats, statmod, VGAM, dplyr, parallel, utils, ggplot2, patchwork, rlang, gridExtra

Suggests knitr, rmarkdown, kableExtra, testthat (>= 3.0.0), purrr, tidyr,

License GPL-3

Encoding UTF-8

RoxygenNote 7.3.3

VignetteBuilder knitr

URL https://rikokelter.github.io/bfbin2arm/, https://arxiv.org/abs/2511.23144, https://arxiv.org/abs/2603.01715, https://arxiv.org/abs/2502.02914, https://arxiv.org/abs/2606.02410

NeedsCompilation no

Author Riko Kelter [aut, cre]

Maintainer Riko Kelter <rkelter@uni-koeln.de>

Repository https://cran.r-universe.dev

Date/Publication 2026-10-28 13:10:09 UTC

RemoteUrl https://github.com/cran/bfbin2arm

RemoteRef HEAD

RemoteSha 11d9aed11e6bd1674ddccf249ea705fa912215db

Contents

.sustained_singlearm_feasibility . . . 3
as.data.frame.singlearm_onestage_bf_design . . . 4
as.data.frame.singlearm_rope_twostage_design . . . 5
BFminus0 . . . 5
BFminus1 . . . 6
BFplus0 . . . 6
BFplus1 . . . 7
BFplusMinus . . . 7
calibrate_bf_ttest . . . 8
check_feasibility_rope . . . 10
coef.singlearm_rope_twostage_design . . . 11
design_singlearm_bf . . . 12
design_singlearm_onestage_bf . . . 14
design_singlearm_onestage_rope . . . 16
design_singlearm_twostage_rope . . . 17
design_singlearm_twostage_rope_equiv . . . 19
design_singlearm_twostage_rope_ni . . . 20
design_singlearm_twostage_rope_sup . . . 20
design_twoarm_onestage_bf . . . 21
design_twoarm_onestage_rope . . . 23
design_twoarm_onestage_rope_directional . . . 26
design_twoarm_onestage_rope_ni . . . 28
design_twoarm_onestage_rope_nisup . . . 29
design_twoarm_onestage_rope_sup . . . 30
design_twoarm_twostage_bf . . . 30
make_analysis_prior_greater . . . 33
make_analysis_prior_less_equal . . . 34
make_analysis_prior_two_sided . . . 34
make_design_prior_normal . . . 35
make_design_prior_point . . . 35
make_point_prior . . . 36
make_prior . . . 36
ntwoarmbinbf01 . . . 37
optimal_onestage_singlearm_bf . . . 40
optimal_onestage_twoarm_bf . . . 42
optimal_twostage_2arm_bf . . . 44
optimal_twostage_singlearm_bf . . . 48
plot.bf_ttest_design . . . 50
plot.bfbin2armrope2armdesign . . . 51
plot.singlearm_bf_design . . . 51

plot.singlearm_onestage_bf_design . . . 52
plot.singlearm_rope_twostage_design . . . 53
plot.twoarm_rope_directional_onestage_design . . . 54
plot_twostage_2arm_bf . . . 54
postProbHminus . . . 55
postProbHplus . . . 56
powerbinbf01_fixed . . . 56
powerbinbf01seq . . . 58
powertwoarmbinbf01 . . . 59
predictiveDensityH0 . . . 61
predictiveDensityH1 . . . 62
predictiveDensityHminus_trunc . . . 62
predictiveDensityHplus_trunc . . . 63
print.singlearm_onestage_bf_design . . . 64
print.singlearm_rope_twostage_design . . . 64
print.summary.singlearm_bf_design . . . 65
print.summary.singlearm_onestage_bf_design . . . 65
print.summary.singlearm_rope_twostage_design . . . 66
priorProbHminus . . . 66
priorProbHplus . . . 67
summary.singlearm_bf_design . . . 67
summary.singlearm_onestage_bf_design . . . 68
summary.singlearm_rope_twostage_design . . . 68
twoarmbinbf_plus0_direct . . . 69
twoarmbinbf01 . . . 70

Index **71**

`.sustained_singlearm_feasibility`

Check sustained feasibility over future n

Description

Given vectors of operating characteristics over n, check whether power, type-I-error, and CE(H0) satisfy their thresholds at n and for at least sustain_n subsequent sample sizes.

Usage

```
.sustained_singlearm_feasibility(
  n_vec,
  power_vec,
  type1_vec,
  ce_vec,
  target_power,
  target_type1,
  target_ce,
  sustain_n
)
```

Arguments

n_vec	Integer vector of sample sizes.
power_vec	Numeric vector of power values (same length as n_vec).
type1_vec	Numeric vector of type-I-error values.
ce_vec	Numeric vector of CE(H0) values (may contain NA).
target_power	Numeric target power.
target_type1	Numeric target type-I-error.
target_ce	Numeric target CE(H0); if <= 0, CE constraint is ignored.
sustain_n	Integer, number of subsequent sample sizes that must also satisfy the constraints.

Value

Logical vector of length length(n_vec): TRUE if n_i and the next sustain_n sample sizes satisfy all active constraints.

---

as.data.frame.singlearm_onestage_bf_design

Convert a one-stage single-arm BF design to a data frame

Description

Convert a one-stage single-arm BF design to a data frame

Usage

```
## S3 method for class 'singlearm_onestage_bf_design'
as.data.frame(x, row.names = NULL, optional = FALSE, ...)
```

Arguments

x	An object of class "singlearm_onestage_bf_design".
row.names	Ignored.
optional	Ignored.
...	Currently unused.

Value

A data frame with the search results.

as.data.frame.singlearm_rope_twostage_design *Convert candidates to a data frame*

Description

Convert candidates to a data frame

Usage

```
## S3 method for class 'singlearm_rope_twostage_design'
as.data.frame(x, row.names = NULL, optional = FALSE, ...)
```

Arguments

x	An object of class "singlearm_rope_twostage_design".
row.names	Ignored.
optional	Ignored.
...	Further arguments (currently ignored).

Value

The candidates data frame.

BFminus0 *Bayes factor BF-0: H- vs H0*

Description

Bayes factor BF-0: H- vs H0

Usage

```
BFminus0(BFminus1, BF01)
```

Arguments

BFminus1	Value of BF_{-1}.
BF01	Value of BF_{01}.

Value

Numeric scalar, BF_{-0}.

BFminus1 *Bayes factor BF-1: H- vs H1*

Description

Bayes factor BF-1: H- vs H1

Usage

```
BFminus1(y1, y2, n1, n2, a_1_a = 1, b_1_a = 1, a_2_a = 1, b_2_a = 1)
```

Arguments

y1	Number of successes in arm 1 (control).
y2	Number of successes in arm 2 (treatment).
n1	Sample size in arm 1.
n2	Sample size in arm 2.
a_1_a, b_1_a	Shape parameters of the Beta prior for p_1 under the alternative (analysis prior).
a_2_a, b_2_a	Shape parameters of the Beta prior for p_2 under the alternative (analysis prior).

Value

Numeric scalar, BF_{-1}.

BFplus0 *Bayes factor BF+0: H+ vs H0*

Description

Bayes factor BF+0: H+ vs H0

Usage

```
BFplus0(BFplus1, BF01)
```

Arguments

BFplus1	Value of BF_{+1}.
BF01	Value of BF_{01}.

Value

Numeric scalar, BF_{+0}.

BFplus1 *Bayes factor BF+1: H+ vs H1*

Description

Bayes factor BF+1: H+ vs H1

Usage

```
BFplus1(y1, y2, n1, n2, a_1_a = 1, b_1_a = 1, a_2_a = 1, b_2_a = 1)
```

Arguments

y1	Number of successes in arm 1 (control).
y2	Number of successes in arm 2 (treatment).
n1	Sample size in arm 1.
n2	Sample size in arm 2.
a_1_a, b_1_a	Shape parameters of the Beta prior for p_1 under the alternative (analysis prior).
a_2_a, b_2_a	Shape parameters of the Beta prior for p_2 under the alternative (analysis prior).

Value

Numeric scalar, BF_{+1} = posterior odds / prior odds for H+ vs H1.

BFplusMinus *Bayes factor BF+-: H+ vs H-*

Description

Bayes factor BF+-: H+ vs H-

Usage

```
BFplusMinus(BFplus1, BFminus1)
```

Arguments

BFplus1	Value of BF_{+1}.
BFminus1	Value of BF_{-1}.

Value

Numeric scalar, BF_{+-}.

`calibrate_bf_ttest` *Calibrate a Bayesian t-test design*

Description

Searches an integer sample-size range and returns the smallest design that satisfies the requested frequentist and Bayesian operating-characteristic constraints for a one-sample or paired Bayesian t-test.

Usage

```
calibrate_bf_ttest(
  design = c("one.sample", "paired", "two.sample.pooled"),
 hypothesis = c("two.sided.pointnull", "greater.pointnull", "greater.composite"),
  n_min,
  n_max,
  k1 = 1/10,
  k0 = 10,
  analysis_prior_sd = 0.5,
  analysis_prior = NULL,
  freq_t1e = NULL,
  freq_power = NULL,
  bayes_power = NULL,
  bayes_t1e = NULL,
  pce_h0 = NULL,
  design_prior_h1 = list(type = "point", effect = 0.4),
  design_prior_h0 = list(type = "point", effect = 0),
  analysis_prior_h0 = NULL,
  analysis_prior_h1 = NULL,
  control = NULL
)
```

Arguments

`design`	Design type. One of `"one.sample"`, `"paired"`, or `"two.sample.pooled"`. The current implementation supports `"one.sample"` and `"paired"`.
`hypothesis`	Hypothesis configuration. One of `"two.sided.pointnull"`, `"greater.pointnull"`, or `"greater.composite"`.
`n_min`	Minimum candidate sample size.
`n_max`	Maximum candidate sample size.
`k1`	Evidence threshold for evidence in favour of the alternative: evidence for H1 is declared when `BF01 <= k1`.
`k0`	Evidence threshold for evidence in favour of the null: evidence for H0 is declared when `BF01 >= k0`.

`analysis_prior_sd`
: Standard deviation of the default analysis prior. Used when a relevant explicit analysis prior is not supplied.

`analysis_prior`
: Analysis prior for point-null hypotheses. If `NULL`, a default prior determined by `hypothesis` and `analysis_prior_sd` is used.

`freq_t1e`
: Optional upper bound on the frequentist type-I error, calculated under the null effect.

`freq_power`
: Optional list of frequentist power specifications. Each element must contain `effect` and may contain a target `target`.

`bayes_power`
: Optional lower bound on Bayesian power, averaged under `design_prior_h1`.

`bayes_t1e`
: Optional upper bound on the Bayesian type-I error, averaged under `design_prior_h0`.

`pce_h0`
: Optional lower bound on the probability of compelling evidence for H0, averaged under `design_prior_h0`.

`design_prior_h1`
: Design-prior specification under H1.

`design_prior_h0`
: Design-prior specification under H0.

`analysis_prior_h0`
: Analysis prior under H0 for `"greater.composite"` hypotheses.

`analysis_prior_h1`
: Analysis prior under H1 for `"greater.composite"` hypotheses.

`control`
: Optional named list controlling numerical integration, root-finding, search strategy, verification, and progress reporting.

Value

An object of class `"bf_ttest_design"`. On success, it contains the selected sample size (`n`), decision regions, operating characteristics, a constraint table, search results, and plot data. If no feasible design is found, it returns a `"bf_ttest_design"` object with `success = FALSE`.

Examples

```
## Not run:
design <- calibrate_bf_ttest(
  design = "one.sample",
  hypothesis = "greater.composite",
  n_min = 20,
  n_max = 120,
  k1 = 1 / 20,
  k0 = 3,
  freq_t1e = 0.05,
  freq_power = list(main = list(effect = 0.40, target = 0.80)),
  design_prior_h0 = list(
    type = "truncnormal",
    mean = -0.10,
    sd = 0.25,
    lower = -Inf,
```

```
    upper = 0
  ),
  design_prior_h1 = list(
    type = "truncnormal",
    mean = 0.40,
    sd = 0.20,
    lower = 0,
    upper = Inf
  ),
  control = list(progress = FALSE)
)

## End(Not run)
```

`check_feasibility_rope` *Check asymptotic feasibility of a ROPE-based design*

Description

Computes the asymptotic (n -> Inf) ceiling on Bayesian predictive power under H1 and the asymptotic floor on Bayesian predictive type-I error under H0, implied by the specified beta design priors and decision boundary. These quantities provide a necessary (but not sufficient) condition for the feasibility of a calibrated design, and can be used to diagnose "No feasible design found" results from `design_singlearm_onestage_rope` before running the full numerical root-finding search. See Lemma 1 (non-inferiority / superiority) and Lemma 2 (equivalence) in the accompanying manuscript.

Usage

```
check_feasibility_rope(
  p0,
  delta,
  da0,
  db0,
  da1,
  db1,
  direction = c("equivalence", "noninferiority", "superiority"),
  target_power = NULL,
  target_type1 = NULL,
  verbose = TRUE
)
```

Arguments

`p0`	Benchmark response probability.
`delta`	ROPE half-width (equivalence), NI margin, or superiority margin.

da0, db0	Design prior parameters under H0, Beta(da0, db0).
da1, db1	Design prior parameters under H1, Beta(da1, db1).
direction	Decision type: "equivalence", "noninferiority", or "superiority".
target_power	Target Bayesian predictive power under H1 (optional).
target_type1	Target Bayesian predictive type-I error under H0 (optional).
verbose	Logical; if TRUE (default), prints a human-readable feasibility report.

Value

A list with components power_ceiling, type1_floor, power_feasible (logical or NA if no target given), and type1_feasible (logical or NA if no target given).

Examples

```
check_feasibility_rope(
  p0 = 0.30, delta = 0.10,
  da0 = 45, db0 = 105, da1 = 44, db1 = 36,
  direction = "superiority",
  target_power = 0.80, target_type1 = 0.10
)
```

---

coef.singlearm_rope_twostage_design

Extract design coefficients

Description

Extract design coefficients

Usage

```
## S3 method for class 'singlearm_rope_twostage_design'
coef(object, ...)
```

Arguments

object	An object of class "singlearm_rope_twostage_design".
...	Further arguments (currently ignored).

Value

A named numeric vector of key design quantities.

`design_singlearm_bf` *Design or evaluate a single-arm two-stage Bayes factor trial*

Description

Calibrates or evaluates a single-arm two-stage Bayes factor design for a binary endpoint with one interim analysis for futility.

Usage

```
design_singlearm_bf(
  n1_min,
  n2_max,
  k,
  k_f,
  p0,
  a0 = 1,
  b0 = 1,
  a1 = 1,
  b1 = 1,
  dp = NA_real_,
  da0 = 1,
  db0 = 1,
  da1 = 1,
  db1 = 1,
  type = c("point", "direction"),
  calibration = c("Bayesian", "frequentist", "hybrid", "full"),
  target_power = 0.8,
  target_type1 = 0.05,
  target_ce_h0 = 0,
  target_freq_power = 0.8,
  target_freq_type1 = 0.05,
  algorithm = c("optimal", "manual"),
  interim = NULL,
  final = NULL,
  power_cushion = 0,
  ...
)
```

Arguments

`n1_min`	Integer. Minimum admissible interim sample size.
`n2_max`	Integer. Maximum admissible final sample size.
`k`	Numeric scalar greater than 0. Efficacy threshold on the BF_{01} scale.
`k_f`	Numeric scalar greater than 1. Futility threshold on the BF_{01} scale.
`p0`	Numeric scalar in $(0, 1)$. Null response probability.

a0, b0	Positive numeric scalars. Beta analysis-prior parameters under H_0.
a1, b1	Positive numeric scalars. Beta analysis-prior parameters under H_1.
dp	Optional numeric scalar in $(0, 1)$. Fixed point alternative used for frequentist power calculations under H_1.
da0, db0	Positive numeric scalars. Beta design-prior parameters under H_0.
da1, db1	Positive numeric scalars. Beta design-prior parameters under H_1.
type	Character string specifying the Bayes-factor test. One of "point" or "direction".
calibration	Character string specifying the calibration mode. One of "Bayesian", "frequentist", "hybrid", or "full".
target_power	Numeric scalar in $(0, 1)$. Target corrected Bayesian power.
target_type1	Numeric scalar in $(0, 1)$. Target corrected Bayesian type-I error.
target_ce_h0	Numeric scalar in $[0, 1)$. Optional lower bound on the corrected Bayesian probability of compelling evidence in favour of H_0.
target_freq_power	Numeric scalar in $(0, 1)$. Target corrected frequentist power at dp.
target_freq_type1	Numeric scalar in $(0, 1)$. Target corrected frequentist type-I error at $p = p_0$.
algorithm	Character string specifying whether the design should be optimized or only evaluated.
interim	Optional integer interim sample size used when algorithm = "manual".
final	Optional integer final sample size used when algorithm = "manual".
power_cushion	Optional additive cushion applied only in the fixed-sample anchor search of the first optimization step. This can be useful because introducing an interim futility analysis typically reduces corrected power relative to the fixed-sample anchor.
...	Reserved for future extensions.

Details

The design uses the Bayes factor BF_{01}. Small values of BF_{01} indicate evidence against H_0, so final efficacy is concluded when BF_{01}
lek. Large values indicate evidence in favour of H_0, so interim futility is concluded when BF_{01}
gek_f.

Analysis priors are specified separately under H_0 and H_1 via a0, b0, a1, b1. Design priors are specified separately under H_0 and H_1 via da0, db0, da1, db1.

Value

An object of class "singlearm_bf_design".

`design_singlearm_onestage_bf` *Design or evaluate a one-stage single-arm Bayes factor trial*

Description

Calibrates or evaluates a one-stage single-arm Bayes factor design for a binary endpoint.

Usage

```
design_singlearm_onestage_bf(
  n_min,
  n_max,
  k,
  k_ce = NULL,
  p0,
  a0 = 1,
  b0 = 1,
  a1 = 1,
  b1 = 1,
  dp = NA_real_,
  da0 = 1,
  db0 = 1,
  da1 = 1,
  db1 = 1,
  type = c("point", "direction"),
  calibration = c("Bayesian", "frequentist", "hybrid", "full"),
  target_power = 0.8,
  target_type1 = 0.05,
  target_ce_h0 = 0,
  target_freq_power = 0.8,
  target_freq_type1 = 0.05,
  algorithm = c("optimal", "manual"),
  n = NULL,
  power_cushion = 0,
  sustain_n = 10L,
  ...
)
```

Arguments

`n_min`	Integer. Minimum admissible sample size.
`n_max`	Integer. Maximum admissible sample size.
`k`	Numeric scalar greater than 0. Evidence threshold on the BF_{01} scale for efficacy, used for power and type-I error.

k_ce	Optional numeric scalar greater than 1. Threshold on the BF_{01} scale used for CE(H0) / PCE(H0). Must be supplied when target_ce_h0 > 0.
p0	Numeric scalar in $(0, 1)$. Null response probability.
a0, b0	Positive numeric scalars. Beta analysis-prior parameters under H_0.
a1, b1	Positive numeric scalars. Beta analysis-prior parameters under H_1.
dp	Optional numeric scalar in $(0, 1)$. Fixed point alternative used for frequentist power calculations under H_1.
da0, db0	Positive numeric scalars. Beta design-prior parameters under H_0.
da1, db1	Positive numeric scalars. Beta design-prior parameters under H_1.
type	Character string specifying the Bayes-factor test. One of "point" or "direction".
calibration	Character string specifying the calibration mode. One of "Bayesian", "frequentist", "hybrid", or "full".
target_power	Numeric scalar in $(0, 1)$. Target corrected Bayesian power.
target_type1	Numeric scalar in $(0, 1)$. Target corrected Bayesian type-I error.
target_ce_h0	Numeric scalar in $[0, 1)$. Optional lower bound on the corrected Bayesian probability of compelling evidence in favour of H_0.
target_freq_power	Numeric scalar in $(0, 1)$. Target corrected frequentist power at dp.
target_freq_type1	Numeric scalar in $(0, 1)$. Target corrected frequentist type-I error at $p = p_0$.
algorithm	Character string specifying whether the design should be optimized or only evaluated.
n	Optional integer sample size used when algorithm = "manual".
power_cushion	Optional additive cushion applied to the power targets in the optimizer.
sustain_n	Non-negative integer. A candidate design is considered feasible only if the relevant operating characteristics satisfy their target constraints at the candidate sample size and for the next sustain_n larger sample sizes, subject to the search range. This also applies to the CE(H0) constraint when target_ce_h0 > 0.
...	Reserved for future extensions.

Details

The design uses the Bayes factor BF_{01}. Small values of BF_{01} indicate evidence against H_0, so efficacy is concluded when $BF_{01} \leq k$. Large values indicate evidence in favour of H_0, and the optional CE(H0) / PCE(H0) constraint is evaluated using the separate threshold k_ce.

Analysis priors are specified separately under H_0 and H_1 via a0, b0, a1, b1. Design priors are specified separately under H_0 and H_1 via da0, db0, da1, db1.

Value

An object of class "singlearm_onestage_bf_design".

`design_singlearm_onestage_rope` *Calibrate a one-stage single-arm ROPE design for a binomial endpoint*

Description

Calibrate a one-stage single-arm ROPE design for a binomial endpoint

Usage

```
design_singlearm_onestage_rope(
  n_min,
  n_max,
  p0,
  delta,
  gamma_eq,
  gamma_diff = gamma_eq,
  direction = c("equivalence", "noninferiority", "superiority"),
  a = 1,
  b = 1,
  da0,
  db0,
  da1,
  db1,
  calibration = c("Bayesian", "frequentist", "hybrid", "full"),
  dp = NULL,
  target_power = NULL,
  target_type1 = NULL,
  target_pce_h0 = NULL,
  target_freq_power = NULL,
  target_freq_type1 = NULL,
  sustain_n = 1,
  return_grid = TRUE
)
```

Arguments

`n_min`	Minimum sample size.
`n_max`	Maximum sample size.
`p0`	Benchmark response probability.
`delta`	ROPE half-width (equivalence), NI margin, or superiority margin.
`gamma_eq`	Posterior probability threshold for accepting H1.
`gamma_diff`	Posterior probability threshold for compelling evidence for H0. Defaults to gamma_eq.

direction	Decision type: "equivalence", "noninferiority", or "superiority".
a, b	Analysis prior parameters for Beta(a,b).
da0, db0	Design prior parameters under H0.
da1, db1	Design prior parameters under H1.
calibration	Calibration mode: "Bayesian", "frequentist", "hybrid", or "full".
dp	Point alternative in the favorable H1 region at which frequentist power is computed.
target_power	Target Bayesian predictive power under H1.
target_type1	Target Bayesian predictive type-I error under H0.
target_pce_h0	Optional target for predictive compelling evidence for H0 under H0.
target_freq_power	Target frequentist power at dp.
target_freq_type1	Target worst-case frequentist type-I error at the null boundary.
sustain_n	Number of consecutive feasible sample sizes required.
return_grid	Return the full evaluation grid.

Value

An object of class bfbin2arm_rope_design.

`design_singlearm_twostage_rope` *Calibrate an optimal single-arm two-stage ROPE design*

Description

Finds a single-arm two-stage Bayesian design based on the region of practical equivalence (ROPE) for a binary endpoint, with a single interim analysis allowing early stopping for futility. The design covers three decision types via the `direction` argument:

`"equivalence"` Posterior mass inside the two-sided ROPE $[p_0 - \delta,\ p_0 + \delta]$ must exceed γ_{eq}.

`"noninferiority"` Posterior probability $\Pr(p \geq p_0 - \delta \mid Y)$ must exceed γ_{eq}.

`"superiority"` Posterior probability $\Pr(p > p_0 + \delta \mid Y)$ must exceed γ_{eq}.

Usage

```
design_singlearm_twostage_rope(
  p0,
  delta,
  analysis_prior = c(1, 1),
  design_prior_h0,
  design_prior_h1,
```

```
  gamma_1 = 0.5,
  gamma_eq = 0.9,
  gamma_diff = gamma_eq,
  alpha = 0.1,
  power = 0.8,
  pce = NULL,
  alpha_freq = NULL,
  power_freq = NULL,
  p_t1e = NULL,
  p_power = NULL,
  nmax = 300L,
  direction = c("equivalence", "noninferiority", "superiority"),
  minimax = FALSE,
  progress = TRUE
)
```

Arguments

`p0`	Benchmark response probability.
`delta`	ROPE half-width (`"equivalence"`), non-inferiority margin (`"noninferiority"`), or superiority margin (`"superiority"`). Must be a single positive number.
`analysis_prior`	Numeric vector `c(a, b)` for the $\mathrm{Beta}(a, b)$ analysis prior on p. Defaults to `c(1, 1)` (uniform).
`design_prior_h0`	Numeric vector `c(a, b)` for the null design prior.
`design_prior_h1`	Numeric vector `c(a, b)` for the alternative design prior.
`gamma_1`	Interim futility threshold in $(0, 1)$ applied to the posterior support for H_0. In the equivalence design, the trial stops early for futility if $\Pr(p \notin \mathcal{R}_p \mid Y_1) \geq \gamma_1$, equivalently if the interim posterior ROPE probability is at most $1 - \gamma_1$. Continuation to stage 2 occurs otherwise.
`gamma_eq`	Final evidence threshold in $(0.5, 1)$: the appropriate posterior ROPE probability must exceed `gamma_eq` to declare equivalence, non-inferiority, or superiority.
`gamma_diff`	Threshold for compelling evidence for H_0: the complementary posterior ROPE probability must exceed `gamma_diff`. In the two-stage design, compelling evidence for H_0 may be obtained either at the interim analysis or at the final analysis. Defaults to `gamma_eq`.
`alpha`	Target predictive type-I error level (upper bound).
`power`	Target predictive power (lower bound).
`pce`	Optional lower bound on predictive `PCE(H0)`.
`alpha_freq`	Optional upper bound on the frequentist type-I error.
`power_freq`	Optional lower bound on the frequentist power.
`p_t1e`	Point at which the frequentist type-I error is evaluated.
`p_power`	Point at which the frequentist power is evaluated.

nmax	Upper bound on the fixed-sample size n^* searched in step 1. An informative error is raised if no feasible size is found.
direction	Character string specifying the decision type. One of "equivalence" (default), "noninferiority", or "superiority".
minimax	Logical. If TRUE, minimise n (minimax criterion); if FALSE (default), minimise EN_0 (optimal criterion).
progress	Logical. If TRUE (default), print progress messages.

Details

The search proceeds in two steps: (1) find the minimum fixed-sample size n^* at which the one-stage constraints are satisfied; (2) enumerate all two-stage splits $n_1 + n_2 = n^*$ and retain those satisfying the two-stage constraints. The optimal design minimises EN_0 (or n^* under the minimax criterion) among all feasible splits.

Value

An object of class "singlearm_rope_twostage_design".

---

design_singlearm_twostage_rope_equiv

Calibrate a two-stage ROPE equivalence design

Description

Convenience wrapper for design_singlearm_twostage_rope() with direction = "equivalence".

Usage

```
design_singlearm_twostage_rope_equiv(...)
```

Arguments

...	Arguments passed to design_singlearm_twostage_rope(). The direction argument is set internally to "equivalence" and must not be supplied.

Value

An object of class "singlearm_rope_twostage_design".

See Also

design_singlearm_twostage_rope() for all supported arguments and details of the design calculation.

design_singlearm_twostage_rope_ni

Calibrate a two-stage ROPE non-inferiority design

Description

Convenience wrapper for `design_singlearm_twostage_rope()` with `direction = "noninferiority"`.

Usage

```
design_singlearm_twostage_rope_ni(...)
```

Arguments

`...`	Arguments passed to `design_singlearm_twostage_rope()`. The `direction` argument is set internally to `"noninferiority"` and must not be supplied.

Value

An object of class `"singlearm_rope_twostage_design"`.

See Also

`design_singlearm_twostage_rope()` for all supported arguments and details of the design calculation.

design_singlearm_twostage_rope_sup

Calibrate a two-stage ROPE superiority design

Description

Convenience wrapper for `design_singlearm_twostage_rope()` with `direction = "superiority"`.

Usage

```
design_singlearm_twostage_rope_sup(...)
```

Arguments

`...`	Arguments passed to `design_singlearm_twostage_rope()`. The `direction` argument is set internally to `"superiority"` and must not be supplied.

Value

An object of class `"singlearm_rope_twostage_design"`.

See Also

design_singlearm_twostage_rope() for all supported arguments and details of the design calculation.

`design_twoarm_onestage_bf` *Design or evaluate a one-stage two-arm Bayes factor trial*

Description

Calibrates or evaluates a one-stage two-arm Bayes factor design for a binary endpoint with fixed randomisation between the two arms.

Usage

```
design_twoarm_onestage_bf(
  n_min,
  n_max,
  k = 1/3,
  k_f = 3,
  test = c("BF01", "BF+0", "BF-0", "BF+-"),
  a_0_d = 1,
  b_0_d = 1,
  a_0_a = 1,
  b_0_a = 1,
  a_1_d = 1,
  b_1_d = 1,
  a_2_d = 1,
  b_2_d = 1,
  a_1_a = 1,
  b_1_a = 1,
  a_2_a = 1,
  b_2_a = 1,
  a_1_d_Hminus = 1,
  b_1_d_Hminus = 1,
  a_2_d_Hminus = 1,
  b_2_d_Hminus = 1,
  a_1_a_Hminus = 1,
  b_1_a_Hminus = 1,
  a_2_a_Hminus = 1,
  b_2_a_Hminus = 1,
  alloc1 = 0.5,
  alloc2 = 0.5,
  calibration = c("Bayesian", "frequentist", "hybrid", "full"),
  target_power = 0.8,
  target_type1 = 0.05,
```

```
  target_ce_h0 = 0,
  target_freq_power = 0.8,
  target_freq_type1 = 0.05,
  p1_grid = seq(0.01, 0.99, 0.02),
  p2_grid = seq(0.01, 0.99, 0.02),
  p1_power = NULL,
  p2_power = NULL,
  power_cushion = 0,
  sustain_n = 10L,
  report_freq_type1 = FALSE,
  algorithm = c("optimal", "manual"),
  n_total = NULL,
  progress = FALSE,
  ...
)
```

Arguments

n_min	Integer. Minimum admissible total sample size.
n_max	Integer. Maximum admissible total sample size.
k	Numeric scalar greater than 0. Evidence threshold used for power and type-I error.
k_f	Numeric scalar greater than 1. Threshold used for CE(H0) / PCE(H0).
test	Character string, one of "BF01", "BF+0", "BF-0", or "BF+-".
a_0_d, b_0_d, a_0_a, b_0_a	Shape parameters for design and analysis priors under H_0.
a_1_d, b_1_d, a_2_d, b_2_d	Shape parameters for design priors under H_1 or H_+.
a_1_a, b_1_a, a_2_a, b_2_a	Shape parameters for analysis priors under H_1 or H_+.
a_1_d_Hminus, b_1_d_Hminus, a_2_d_Hminus, b_2_d_Hminus	Optional design priors under H_- for directional tests.
a_1_a_Hminus, b_1_a_Hminus, a_2_a_Hminus, b_2_a_Hminus	Optional analysis priors under H_- for directional tests.
alloc1, alloc2	Fixed randomisation probabilities for arm 1 and arm 2. Must be positive and sum to 1.
calibration	Character string specifying the calibration mode. One of "Bayesian", "frequentist", "hybrid", or "full".
target_power	Numeric scalar in $(0, 1)$. Target corrected Bayesian power.
target_type1	Numeric scalar in $(0, 1)$. Target corrected Bayesian type-I error.
target_ce_h0	Numeric scalar in $[0, 1)$. Optional lower bound on the corrected Bayesian probability of compelling evidence in favour of H_0 (or H_- for test = "BF+-").
target_freq_power	Numeric scalar in $(0, 1)$. Target frequentist power under p1_power, p2_power.

`target_freq_type1`
: Numeric scalar in $(0, 1)$. Target frequentist type-I error.

`p1_grid`, `p2_grid`
: Grids of true proportions used to compute supremum frequentist type-I error.

`p1_power`, `p2_power`
: Optional true proportions used for frequentist power.

`power_cushion`
: Non-negative numeric scalar. Optional additive cushion applied to the power targets during calibration.

`sustain_n`
: Non-negative integer. A candidate total sample size is considered feasible only if the relevant target constraints hold at that total sample size and for the next `sustain_n` larger total sample sizes in the search range.

`report_freq_type1`
: Logical. If `TRUE`, compute and report the frequentist type-I error for the final selected design even when the chosen calibration mode does not use frequentist criteria. This additional computation has no effect on the calibration itself. Defaults to `FALSE`.

`algorithm`
: Character string specifying whether the design should be optimized or only evaluated.

`n_total`
: Optional integer total sample size used when `algorithm = "manual"`.

`progress`
: Logical; if `TRUE`, print simple progress information during optimization.

`...`
: Reserved for future extensions.

Details

The design uses one of the Bayes factor tests implemented in `powertwoarmbinbf01()`. Small values of the relevant inverted Bayes factor indicate evidence against the null, so efficacy is concluded when the Bayes factor is below `k`. Large values indicate evidence in favour of the null (or H_- for `test = "BF+-"`), and the optional CE(H0) / PCE(H0) constraint is evaluated using `k_f`.

Value

An object of class `"twoarm_onestage_bf_design"`.

`design_twoarm_onestage_rope` *Calibrate a one-stage two-arm ROPE equivalence design for binary endpoints*

Description

Calculates equal-allocation sample sizes for a parallel-group one-stage equivalence study with binary endpoints. The treatment effect is the risk difference `pT - pC`. Equivalence is declared when the posterior probability of `abs(pT - pC) < delta` is at least `gammaeq`; meaningful non-equivalence is declared when the posterior probability outside the ROPE is at least `gammadiff`; otherwise the decision is inconclusive.

Usage

```
design_twoarm_onestage_rope(
  nmin,
  nmax,
  nstep = 1L,
  delta,
  gammaeq = 0.9,
  gammadiff = gammaeq,
  analysis_prior_C = c(1, 1),
  analysis_prior_T = c(1, 1),
  design_prior_eq_C = NULL,
  design_prior_eq_T = NULL,
  design_prior_eq = NULL,
  design_prior_ne_C,
  design_prior_ne_T,
  targetpower = NULL,
  targettype1 = NULL,
  targetpce_h0 = NULL,
  sustainn = 1L,
  rel.tol = 1e-10,
  integration = c("adaptive", "quantile"),
  quad_nodes = 128L,
  calibration = c("Bayesian", "frequentist", "hybrid", "full"),
  freq_pC = NULL,
  freq_delta = NULL,
  targetfreqpower = NULL,
  targetfreqtype1 = NULL,
  freq_grid_n = 101L,
  freq_pC_range = c(0, 1),
  parallel = FALSE,
  ncores = 1L,
  parallel_backend = c("auto", "psock", "multicore", "none"),
  returngrid = TRUE,
  returnmatrices = FALSE,
  progress = interactive(),
  progress_every = 1L
)
```

Arguments

nmin, nmax	Minimum and maximum per-arm sample size.
nstep	Positive per-arm sample-size increment.
delta	ROPE half-width on the risk-difference scale.
gammaeq	Posterior threshold for equivalence.
gammadiff	Posterior threshold for meaningful non-equivalence.
analysis_prior_C, analysis_prior_T	Length-two Beta analysis-prior shapes.

`design_prior_eq_C`, `design_prior_eq_T`
: Legacy equivalence-prior Beta kernels.

`design_prior_eq`
: Optional joint baseline-risk/risk-difference design prior.

`design_prior_ne_C`, `design_prior_ne_T`
: Non-equivalence-prior Beta kernels.

`targetpower`
: Target Bayesian predictive probability of declaring practical equivalence under the equivalence design distribution `G1`.

`targettype1`
: Maximum Bayesian predictive probability of falsely declaring practical equivalence under the non-equivalence design distribution `G0`.

`targetpce_h0`
: Optional minimum Bayesian predictive probability of compelling evidence for meaningful non-equivalence under `G0`. More precisely, this is the predictive probability of declaring meaningful difference, i.e. the posterior probability of `abs(pT - pC) >= delta` is at least `gammadiff`. If supplied, this criterion is enforced for `"Bayesian"`, `"hybrid"`, and `"full"` calibration.

`sustainn`
: Number of consecutive feasible evaluated sample sizes.

`rel.tol`
: Adaptive-integration relative tolerance.

`integration`
: Either `"adaptive"` or `"quantile"`.

`quad_nodes`
: Number of Gauss-Legendre nodes for quantile quadrature.

`calibration`
: One of `"Bayesian"`, `"frequentist"`, `"hybrid"`, or `"full"`.

`freq_pC`
: Control response probability for frequentist pointwise power.

`freq_delta`
: Treatment-control difference for frequentist pointwise power.

`targetfreqpower`
: Target pointwise frequentist equivalence power.

`targetfreqtype1`
: Maximum boundary-grid frequentist type-I error.

`freq_grid_n`
: Boundary-grid size.

`freq_pC_range`
: Control-risk interval for boundary-grid evaluation.

`parallel`
: Logical; if `TRUE`, evaluate candidate per-arm sample sizes in parallel.

`ncores`
: Number of workers if `parallel = TRUE`.

`parallel_backend`
: Parallel backend. `"auto"` and `"psock"` use a PSOCK cluster, which is supported on Windows, macOS, and Linux. `"none"` evaluates candidates sequentially. `"multicore"` is retained for compatibility but currently resolves to `"psock"` because forked processes can be unsafe with threaded numerical libraries.

`returngrid`
: Return the full operating-characteristic grid.

`returnmatrices`
: Retain matrices for the selected design.

`progress`
: Logical; if `TRUE`, show a progress bar during sequential evaluation. Progress display is disabled for parallel evaluation.

`progress_every`
: Positive integer specifying the interval, in evaluated candidate sample sizes, at which sequential progress is updated.

Value

An object of class `"bfbin2armrope2armdesign"`.

`design_twoarm_onestage_rope_directional` *Calibrate a one-stage two-arm directional ROPE design for binary endpoints*

Description

Calculates equal-allocation sample sizes for a parallel-group one-stage study with binary endpoints, where the treatment effect is the risk difference `pT - pC`. Depending on `direction`, the design targets non-inferiority, superiority, or a nested procedure in which superiority is assessed only after non-inferiority has been established. Superiority is declared when the posterior probability of `pT - pC > delta_sup` is at least `gamma_sup`; non-inferiority is declared when the posterior probability of `pT - pC > -delta_ni` is at least `gamma_ni`; harmful inferiority is declared when the posterior probability of `pT - pC <= -delta_ni` is at least `gamma_inf`; otherwise the decision is inconclusive.

Usage

```
design_twoarm_onestage_rope_directional(
  nmin,
  nmax,
  nstep = 1L,
  direction = c("noninferiority", "superiority", "nested"),
  delta_ni,
  delta_sup = 0,
  gamma_ni = 0.975,
  gamma_sup = 0.975,
  gamma_inf = 0.975,
  analysis_prior_C = c(1, 1),
  analysis_prior_T = c(1, 1),
  design_prior_inf = NULL,
  design_prior_ni = NULL,
  design_prior_sup = NULL,
  design_prior_not_sup = NULL,
  target_ni_power = NULL,
  target_sup_power = NULL,
  target_false_ni = NULL,
  target_false_sup = NULL,
  sustainn = 1L,
  rel.tol = 1e-10,
  integration = c("adaptive", "quantile"),
  quad_nodes = 128L,
  calibration = c("Bayesian", "frequentist", "hybrid", "full"),
  freq_ni_pC = NULL,
  freq_ni_delta = NULL,
  freq_sup_pC = NULL,
  freq_sup_delta = NULL,
  target_freq_ni_power = NULL,
```

```
  target_freq_sup_power = NULL,
  target_freq_false_ni = NULL,
  target_freq_false_sup = NULL,
  freq_grid_n = 101L,
  freq_pC_range = c(0, 1),
  parallel = FALSE,
  ncores = 1L,
  parallel_backend = c("auto", "psock", "none"),
  returngrid = TRUE,
  returnmatrices = FALSE,
  progress = interactive(),
  progress_every = 1L
)
```

Arguments

`nmin`, `nmax`	Minimum and maximum per-arm sample size.
`nstep`	Positive per-arm sample-size increment.
`direction`	One of `"noninferiority"`, `"superiority"`, or `"nested"`.
`delta_ni`	Non-inferiority margin on the risk-difference scale.
`delta_sup`	Superiority margin on the risk-difference scale, `>= 0` by convention, with `delta_sup > -delta_ni`.
`gamma_ni`	Posterior threshold for non-inferiority.
`gamma_sup`	Posterior threshold for superiority; must be at least `gamma_ni`.
`gamma_inf`	Posterior threshold for harmful inferiority.
`analysis_prior_C`, `analysis_prior_T`	Length-two Beta analysis-prior shapes.
`design_prior_inf`, `design_prior_ni`, `design_prior_sup`, `design_prior_not_sup`	Baseline-risk/risk-difference design priors under the harmful-inferiority, non-inferiority, superiority, and not-superiority design distributions, respectively. Required depending on `direction` and the requested targets.
`target_ni_power`	Target predictive probability of a non-inferiority or superiority conclusion, evaluated under `design_prior_ni`.
`target_sup_power`	Target predictive probability of a superiority conclusion, evaluated under `design_prior_sup`.
`target_false_ni`	Maximum predictive probability of a non-inferiority or superiority conclusion, evaluated under `design_prior_inf`.
`target_false_sup`	Maximum predictive probability of a superiority conclusion, evaluated under `design_prior_not_sup`.
`sustainn`	Number of consecutive feasible evaluated sample sizes.
`rel.tol`	Adaptive-integration relative tolerance.

integration	Either "adaptive" or "quantile".
quad_nodes	Number of Gauss-Legendre nodes for quantile quadrature.
calibration	One of "Bayesian", "frequentist", "hybrid", or "full".
freq_ni_pC, freq_ni_delta	Control response and true difference for pointwise frequentist non-inferiority power.
freq_sup_pC, freq_sup_delta	Control response and true difference for pointwise frequentist superiority power.
target_freq_ni_power, target_freq_sup_power	Target pointwise frequentist powers.
target_freq_false_ni, target_freq_false_sup	Maximum boundary-grid frequentist error rates.
freq_grid_n	Boundary-grid size.
freq_pC_range	Control-risk interval for boundary-grid evaluation.
parallel	Logical; if TRUE, evaluate candidate per-arm sample sizes in parallel.
ncores	Positive integer giving the maximum number of worker processes used when parallel = TRUE.
parallel_backend	Parallel backend. "auto" and "psock" use a PSOCK cluster. "none" evaluates sequentially.
returngrid	Return the full operating-characteristic grid.
returnmatrices	Retain matrices for the selected design.
progress	Logical; if TRUE, show a progress bar during sequential evaluation.
progress_every	Positive integer specifying the interval, in evaluated candidate sample sizes, at which sequential progress is updated.

Value

An object of class "twoarm_rope_directional_onestage_design".

`design_twoarm_onestage_rope_ni` *Calibrate a one-stage two-arm ROPE non-inferiority design*

Description

Convenience wrapper for `design_twoarm_onestage_rope_directional()` with `direction = "noninferiority"`.

Usage

```
design_twoarm_onestage_rope_ni(...)
```

Arguments

...	Arguments passed to design_twoarm_onestage_rope_directional(). The direction argument is set internally to "noninferiority" and must not be supplied.

Value

An object of class "twoarm_rope_directional_onestage_design".

See Also

design_twoarm_onestage_rope_directional() for all supported arguments and details of the design calculation.

---

design_twoarm_onestage_rope_nisup

Calibrate a one-stage two-arm nested non-inferiority/superiority ROPE design

---

Description

Convenience wrapper for design_twoarm_onestage_rope_directional() with direction = "nested". Superiority is only assessed within the non-inferiority acceptance region, so that a superiority conclusion always also satisfies the non-inferiority evidence requirement. This reproduces the hierarchical testing logic used in trials such as SINGLE, where a design may be powered chiefly for non-inferiority while superiority is evaluated as a prespecified secondary claim.

Usage

```
design_twoarm_onestage_rope_nisup(...)
```

Arguments

...	Arguments passed to design_twoarm_onestage_rope_directional(). The direction argument is set internally to "nested" and must not be supplied.

Value

An object of class "twoarm_rope_directional_onestage_design".

See Also

design_twoarm_onestage_rope_directional() for all supported arguments and details of the design calculation.

design_twoarm_onestage_rope_sup

Calibrate a one-stage two-arm ROPE superiority design

Description

Convenience wrapper for `design_twoarm_onestage_rope_directional()` with `direction = "superiority"`.

Usage

```
design_twoarm_onestage_rope_sup(...)
```

Arguments

`...`	Arguments passed to `design_twoarm_onestage_rope_directional()`. The `direction` argument is set internally to `"superiority"` and must not be supplied.

Value

An object of class `"twoarm_rope_directional_onestage_design"`.

See Also

`design_twoarm_onestage_rope_directional()` for all supported arguments and details of the design calculation.

design_twoarm_twostage_bf

Design an optimal two-stage two-arm Bayes factor trial

Description

Calibrates a two-stage two-arm Bayes factor design for a binary endpoint by calling `optimal_twostage_2arm_bf()` and packaging the result in a user-friendly object of class `"twoarm_twostage_bf_design"`.

Usage

```
design_twoarm_twostage_bf(
  n1_min,
  n2_max,
  alloc1 = 0.5,
  alloc2 = 0.5,
  power_cushion = 0,
  interim_fraction = c(0.25, 0.75),
  grid_step = 1L,
```

```
  coarse_step = 4L,
  max_iter = 40L,
  ncores = getOption("bfbin2arm.ncores", 1L),
  k = 1/3,
  k_f = 3,
  test = c("BF01", "BF+0", "BF-0", "BF+-"),
  a_0_d = 1,
  b_0_d = 1,
  a_0_a = 1,
  b_0_a = 1,
  a_1_d = 1,
  b_1_d = 1,
  a_2_d = 1,
  b_2_d = 1,
  a_1_a = 1,
  b_1_a = 1,
  a_2_a = 1,
  b_2_a = 1,
  a_1_d_Hminus = 1,
  b_1_d_Hminus = 1,
  a_2_d_Hminus = 1,
  b_2_d_Hminus = 1,
  a_1_a_Hminus = 1,
  b_1_a_Hminus = 1,
  a_2_a_Hminus = 1,
  b_2_a_Hminus = 1,
  calibration = c("Bayesian", "frequentist", "hybrid"),
  calibration_en = c("Bayesian", "frequentist"),
  target_power = 0.8,
  target_type1 = 0.05,
  target_ce_h0 = 0,
  target_freq_power = 0.8,
  target_freq_type1 = 0.05,
  p1_power = NULL,
  p2_power = NULL,
  p1_en_h0 = NULL,
  p2_en_h0 = NULL,
  p_null_grid = NULL,
  progress = FALSE,
  ...
)
```

Arguments

n1_min	Numeric vector of length 2, minimum interim sample sizes for arms 1 and 2.
n2_max	Numeric vector of length 2, maximum final sample sizes for arms 1 and 2.
alloc1, alloc2	Positive numbers, allocation probabilities to arms 1 and 2.

power_cushion Numeric scalar, optional extra power cushion used in the fixed-sample search of step 1.

interim_fraction
Numeric vector of length 2 giving lower and upper bounds for the interim sample size in each arm as a fraction of the fixed sample size.

grid_step Positive integer giving the spacing of the interim design grid.

coarse_step Positive integer giving the spacing of the coarse fixed-sample search grid in step 1.

max_iter Integer, maximum number of total fixed-sample sizes searched in step 1.

ncores Integer; number of parallel worker processes to use in the calibration. Defaults to `getOption("bfbin2arm.ncores", 1L)`. In vignettes and examples, a conservative value (e.g. 1 or 2) is recommended for CRAN checks, whereas users can increase this to exploit all available cores on their own machines.

k Numeric scalar, efficacy threshold; evidence against the null hypothesis is declared when the corresponding Bayes factor is smaller than `k`.

k_f Numeric scalar, futility threshold; compelling evidence for the null hypothesis is declared when the corresponding Bayes factor is at least `k_f`.

test Character string, one of `"BF01"`, `"BF+0"`, `"BF-0"`, `"BF+-"`.

a_0_d, b_0_d, a_0_a, b_0_a
Shape parameters for design and analysis priors under H_0.

a_1_d, b_1_d, a_2_d, b_2_d
Shape parameters for design priors under H_1 or H_+.

a_1_a, b_1_a, a_2_a, b_2_a
Shape parameters for analysis priors under H_1 or H_+.

a_1_d_Hminus, b_1_d_Hminus, a_2_d_Hminus, b_2_d_Hminus
Optional design priors under H_- for directional tests.

a_1_a_Hminus, b_1_a_Hminus
Shape parameters of the analysis prior under the directional null hypothesis H0- for arm 1.

a_2_a_Hminus, b_2_a_Hminus
Shape parameters of the analysis prior under the directional null hypothesis H0- for arm 2.

calibration Character string specifying the calibration mode at the wrapper level. One of `"Bayesian"`, `"frequentist"`, or `"hybrid"`. This is passed to `optimal_twostage_2arm_bf()` as `calibration_mode`.

calibration_en Character string or `NULL` specifying whether the design is ranked by Bayesian or frequentist expected sample size under the null hypothesis. This is passed to `optimal_twostage_2arm_bf()` as `calibration_EN`.

target_power, target_type1, target_ce_h0, target_freq_power, target_freq_type1
Numeric targets for Bayesian and frequentist operating characteristics. These are translated to the `alpha`, `beta`, `alpha_freq`, and `beta_freq` arguments of `optimal_twostage_2arm_bf()`.

p1_power, p2_power
: Optional true response probabilities used for frequentist power. Passed through to optimal_twostage_2arm_bf().

p1_en_h0, p2_en_h0
: Optional null response probabilities used when calibration_en = "frequentist" to compute expected sample size under the null.

p_null_grid
: Optional grid of null response probabilities used for frequentist type-I-error maximisation.

progress
: Logical; if TRUE, print simple progress information during the calibration.

...
: Reserved for future extensions; currently ignored.

Details

The design uses one of the Bayes factor tests implemented in powertwoarmbinbf01(). Small values of the relevant inverted Bayes factor indicate evidence against the null, so efficacy is concluded when the Bayes factor is below k. Large values indicate evidence in favour of the null (or H_- for test = "BF+-"), and the optional CE(H0) / PCE(H0) constraint is evaluated using k_f.

Value

An object of class "twoarm_twostage_bf_design".

---

make_analysis_prior_greater

Create the default positive-direction analysis prior

Description

Constructs a zero-centred normal prior for the standardized effect, truncated to positive effects. This is the default alternative prior for hypothesis = "greater.pointnull" and the H1 prior for hypothesis = "greater.composite".

Usage

```
make_analysis_prior_greater(sd = 0.5)
```

Arguments

sd
: Positive prior standard deviation.

Value

An object of class "bf_ttest_prior" with support c(0, Inf).

Examples

```
make_analysis_prior_greater(sd = 0.5)
```

make_analysis_prior_less_equal

Create the default non-positive analysis prior

Description

Constructs a zero-centred normal prior for the standardized effect, truncated to non-positive effects. This is the default H0 prior for `hypothesis = "greater.composite"`.

Usage

```
make_analysis_prior_less_equal(sd = 0.5)
```

Arguments

`sd`	Positive prior standard deviation.

Value

An object of class `"bf_ttest_prior"` with support `c(-Inf, 0)`.

Examples

```
make_analysis_prior_less_equal(sd = 0.5)
```

make_analysis_prior_two_sided

Create the default two-sided analysis prior

Description

Constructs a zero-centred normal prior for the standardized effect with support on the entire real line.

Usage

```
make_analysis_prior_two_sided(sd = 0.5)
```

Arguments

`sd`	Positive prior standard deviation.

Value

An object of class `"bf_ttest_prior"`.

Examples

```
make_analysis_prior_two_sided()
```

---

make_design_prior_normal

Create a zero-centred normal design-prior object

Description

Convenience wrapper around `make_prior()` for a normal prior on the standardized effect, with support on the entire real line.

Usage

```
make_design_prior_normal(sd = 0.5)
```

Arguments

`sd`	Positive prior standard deviation.

Value

An object of class `"bf_ttest_prior"`.

Examples

```
make_design_prior_normal(sd = 0.5)
```

---

make_design_prior_point

Backward-compatible alias for `make_point_prior()`

Description

Backward-compatible alias for `make_point_prior()`

Usage

```
make_design_prior_point(effect = 0)
```

Arguments

`effect`	Point effect value.

Value

Object of class `bf_ttest_point_prior`.

make_point_prior *Point-mass prior at a fixed effect*

Description

Point-mass prior at a fixed effect

Usage

```
make_point_prior(effect = 0)
```

Arguments

effect	Scalar effect value.

Value

Object of class bf_ttest_point_prior.

make_prior *Create a prior for the standardized effect*

Description

Create a prior for the standardized effect

Usage

```
make_prior(
  family = c("normal", "student", "cauchy", "jzs", "custom"),
  location = 0,
  scale = 1,
  df = 1,
  support = c(-Inf, Inf),
  log_density = NULL,
  density = NULL,
  name = NULL
)
```

Arguments

family	Prior family.
location	Location parameter.
scale	Scale parameter (> 0).
df	Degrees of freedom for Student prior.
support	Numeric vector c(lower, upper) defining support.
log_density	Optional custom log-density.
density	Optional custom density.
name	Optional display name.

Value

Object of class bf_ttest_prior.

---

ntwoarmbinbf01 *Sample size calibration for two-arm binomial Bayes factor designs*

Description

Searches over a grid of total sample sizes n to find the smallest n such that Bayesian power, Bayesian type-I error, and probability of compelling evidence under H0 meet specified design criteria. Optionally, frequentist type-I error and power constraints are also evaluated. Unequal fixed randomisation between the two arms is allowed via alloc1 and alloc2.

Backward-compatible wrapper around design_twoarm_onestage_bf().

Usage

```
ntwoarmbinbf01(
  k = 1/3,
  k_f = 3,
  power = 0.8,
  alpha = 0.05,
  pce_H0 = 0.9,
  test = c("BF01", "BF+0", "BF-0", "BF+-"),
  nrange = c(10, 150),
  n_step = 1,
  progress = TRUE,
  compute_freq_t1e = FALSE,
  p1_grid = seq(0.01, 0.99, 0.02),
  p2_grid = seq(0.01, 0.99, 0.02),
  p1_power = NULL,
  p2_power = NULL,
  a_0_d = 1,
  b_0_d = 1,
```

```
  a_0_a = 1,
  b_0_a = 1,
  a_1_d = 1,
  b_1_d = 1,
  a_2_d = 1,
  b_2_d = 1,
  a_1_a = 1,
  b_1_a = 1,
  a_2_a = 1,
  b_2_a = 1,
  output = c("plot", "numeric"),
  a_1_d_Hminus = 1,
  b_1_d_Hminus = 1,
  a_2_d_Hminus = 1,
  b_2_d_Hminus = 1,
  a_1_a_Hminus = 1,
  b_1_a_Hminus = 1,
  a_2_a_Hminus = 1,
  b_2_a_Hminus = 1,
  alloc1 = 0.5,
  alloc2 = 0.5,
  sustain_n = 10L
)

ntwoarmbinbf01(
  k = 1/3,
  k_f = 3,
  power = 0.8,
  alpha = 0.05,
  pce_H0 = 0.9,
  test = c("BF01", "BF+0", "BF-0", "BF+-"),
  nrange = c(10, 150),
  n_step = 1,
  progress = TRUE,
  compute_freq_t1e = FALSE,
  p1_grid = seq(0.01, 0.99, 0.02),
  p2_grid = seq(0.01, 0.99, 0.02),
  p1_power = NULL,
  p2_power = NULL,
  a_0_d = 1,
  b_0_d = 1,
  a_0_a = 1,
  b_0_a = 1,
  a_1_d = 1,
  b_1_d = 1,
  a_2_d = 1,
  b_2_d = 1,
  a_1_a = 1,
```

```
  b_1_a = 1,
  a_2_a = 1,
  b_2_a = 1,
  output = c("plot", "numeric"),
  a_1_d_Hminus = 1,
  b_1_d_Hminus = 1,
  a_2_d_Hminus = 1,
  b_2_d_Hminus = 1,
  a_1_a_Hminus = 1,
  b_1_a_Hminus = 1,
  a_2_a_Hminus = 1,
  b_2_a_Hminus = 1,
  alloc1 = 0.5,
  alloc2 = 0.5,
  sustain_n = 10L
)
```

Arguments

k	Evidence threshold for rejecting the null (inverted BF).
k_f	Evidence threshold for "compelling evidence" in favour of the null.
power	Target Bayesian power.
alpha	Target Bayesian type-I error.
pce_H0	Target probability of compelling evidence under H_0.
test	Character string, one of "BF01", "BF+0", "BF-0", "BF+-".
nrange	Integer vector of length 2 giving the search range for total n.
n_step	Step size for n. Currently only n_step = 1 is supported in the object-based calibration workflow.
progress	Logical; if TRUE, print progress to the console.
compute_freq_t1e	Logical; if TRUE, compute frequentist type-I error over a grid.
p1_grid, p2_grid	Grids of true proportions for frequentist T1E.
p1_power, p2_power	Optional true proportions for frequentist power.
a_0_d, b_0_d, a_0_a, b_0_a	Shape parameters for design and analysis priors under H_0.
a_1_d, b_1_d, a_2_d, b_2_d	Shape parameters for design priors under H_1 or H_+.
a_1_a, b_1_a, a_2_a, b_2_a	Shape parameters for analysis priors under H_1 or H_+.
output	"plot" or "numeric".
a_1_d_Hminus, b_1_d_Hminus, a_2_d_Hminus, b_2_d_Hminus	Optional design priors under H_- for directional tests.

`a_1_a_Hminus`, `b_1_a_Hminus`, `a_2_a_Hminus`, `b_2_a_Hminus`
: Shape parameters for analysis priors under H-.

`alloc1`, `alloc2`
: Fixed randomisation probabilities for arm 1 and arm 2; must be positive and sum to 1.

`sustain_n`
: Non-negative integer. A candidate total sample size is considered feasible only if the relevant target constraints hold at that total sample size and for the next `sustain_n` larger total sample sizes in the search range.

Value

If `output = "plot"`, returns invisibly a list with recommended sample sizes and a ggplot object printed to the device. If `output = "numeric"`, returns a list with recommended n and summary.

If `output = "numeric"`, returns a `"twoarm_onestage_bf_design"` object. If `output = "plot"`, the plot is printed and the design object is returned invisibly.

Examples

```
# Standard calibration with equal allocation: power 80%, type-I 5%, CE(H0) 80%

ntwoarmbinbf01(power = 0.8, alpha = 0.05, pce_H0 = 0.8, output = "numeric")


# 1:2 allocation (control:treatment) via alloc1 = 1/3, alloc2 = 2/3

ntwoarmbinbf01(power = 0.8, alpha = 0.05, pce_H0 = 0.8,
               alloc1 = 1/3, alloc2 = 2/3, output = "numeric")


# BF+0 directional test with plot

ntwoarmbinbf01(power = 0.8, alpha = 0.05, pce_H0 = 0.9,
               test = "BF+0", output = "plot")
```

---

optimal_onestage_singlearm_bf

Internal calibration routine for one-stage single-arm BF designs

---

Description

Internal calibration routine for one-stage single-arm BF designs

Usage

```
optimal_onestage_singlearm_bf(
  n_min,
  n_max,
  k,
```

```
  p0,
  a0 = 1,
  b0 = 1,
  a1 = 1,
  b1 = 1,
  dp = NA_real_,
  da0 = 1,
  db0 = 1,
  da1 = 1,
  db1 = 1,
  type = c("point", "direction"),
  calibration = c("Bayesian", "frequentist", "hybrid", "full"),
  target_power = 0.8,
  target_type1 = 0.05,
  target_ce_h0 = 0,
  target_freq_power = 0.8,
  target_freq_type1 = 0.05,
  power_cushion = 0,
  k_ce = NULL,
  sustain_n = 10L
)
```

Arguments

`n_min`	Integer. Minimum admissible sample size in the search grid.
`n_max`	Integer. Maximum admissible sample size in the search grid.
`k`	Numeric scalar greater than 0. Evidence threshold on the BF_{01} scale used for efficacy.
`p0`	Numeric scalar in $(0, 1)$. Null response probability.
`a0`, `b0`	Positive numeric scalars. Beta analysis-prior parameters under H_0.
`a1`, `b1`	Positive numeric scalars. Beta analysis-prior parameters under H_1.
`dp`	Optional numeric scalar in $(0, 1)$. Fixed point alternative used for frequentist power calculations under H_1.
`da0`, `db0`	Positive numeric scalars. Beta design-prior parameters under H_0.
`da1`, `db1`	Positive numeric scalars. Beta design-prior parameters under H_1.
`type`	Character string specifying the Bayes-factor test. One of `"point"` or `"direction"`.
`calibration`	Character string specifying the calibration mode. One of `"Bayesian"`, `"frequentist"`, `"hybrid"`, or `"full"`.
`target_power`	Numeric scalar in $(0, 1)$. Target corrected Bayesian power.
`target_type1`	Numeric scalar in $(0, 1)$. Target corrected Bayesian type-I error.
`target_ce_h0`	Numeric scalar in $[0, 1)$. Optional lower bound on the corrected Bayesian probability of compelling evidence in favour of H_0.
`target_freq_power`	Numeric scalar in $(0, 1)$. Target corrected frequentist power at `dp`.

`target_freq_type1`
: Numeric scalar in $(0, 1)$. Target corrected frequentist type-I error at $p = p_0$.

`power_cushion`
: Non-negative numeric scalar. Optional additive cushion applied to the power targets during calibration.

`k_ce`
: Optional numeric scalar greater than 1. Threshold on the BF_{01} scale used for CE(H0) / PCE(H0) calculations.

`sustain_n`
: Non-negative integer. A candidate sample size is declared feasible only if the relevant constraints are satisfied at that sample size and for the next `sustain_n` larger sample sizes, subject to the search range.

Value

A list with feasibility, selected design, operating characteristics, and full search results.

`optimal_onestage_twoarm_bf` *Internal calibration routine for one-stage two-arm BF designs*

Description

Internal calibration routine for one-stage two-arm BF designs

Usage

```
optimal_onestage_twoarm_bf(
  n_min,
  n_max,
  k = 1/3,
  k_f = 3,
  test = c("BF01", "BF+0", "BF-0", "BF+-"),
  a_0_d = 1,
  b_0_d = 1,
  a_0_a = 1,
  b_0_a = 1,
  a_1_d = 1,
  b_1_d = 1,
  a_2_d = 1,
  b_2_d = 1,
  a_1_a = 1,
  b_1_a = 1,
  a_2_a = 1,
  b_2_a = 1,
  a_1_d_Hminus = 1,
  b_1_d_Hminus = 1,
  a_2_d_Hminus = 1,
  b_2_d_Hminus = 1,
```

```
  a_1_a_Hminus = 1,
  b_1_a_Hminus = 1,
  a_2_a_Hminus = 1,
  b_2_a_Hminus = 1,
  alloc1 = 0.5,
  alloc2 = 0.5,
  calibration = c("Bayesian", "frequentist", "hybrid", "full"),
  target_power = 0.8,
  target_type1 = 0.05,
  target_ce_h0 = 0,
  target_freq_power = 0.8,
  target_freq_type1 = 0.05,
  p1_grid = seq(0.01, 0.99, 0.02),
  p2_grid = seq(0.01, 0.99, 0.02),
  p1_power = NULL,
  p2_power = NULL,
  power_cushion = 0,
  sustain_n = 10L,
  progress = FALSE
)
```

Arguments

`n_min`	Integer. Minimum admissible total sample size.
`n_max`	Integer. Maximum admissible total sample size.
`k`	Numeric scalar greater than 0. Evidence threshold used for power and type-I error.
`k_f`	Numeric scalar greater than 1. Threshold used for CE(H0).
`test`	Character string, one of `"BF01"`, `"BF+0"`, `"BF-0"`, or `"BF+-"`.
`a_0_d`, `b_0_d`, `a_0_a`, `b_0_a`	Shape parameters for design and analysis priors under H_0.
`a_1_d`, `b_1_d`, `a_2_d`, `b_2_d`	Shape parameters for design priors under H_1 or H_+.
`a_1_a`, `b_1_a`, `a_2_a`, `b_2_a`	Shape parameters for analysis priors under H_1 or H_+.
`a_1_d_Hminus`, `b_1_d_Hminus`, `a_2_d_Hminus`, `b_2_d_Hminus`	Optional design priors under H_-.
`a_1_a_Hminus`, `b_1_a_Hminus`, `a_2_a_Hminus`, `b_2_a_Hminus`	Optional analysis priors under H_-.
`alloc1`, `alloc2`	Fixed randomisation probabilities for arm 1 and arm 2.
`calibration`	Character string specifying the calibration mode.
`target_power`, `target_type1`, `target_ce_h0`, `target_freq_power`, `target_freq_type1`	Target operating characteristics.
`p1_grid`, `p2_grid`	Grids for supremum frequentist type-I error.

p1_power, p2_power
: Optional true proportions for frequentist power.

power_cushion
: Non-negative numeric scalar applied to power targets.

sustain_n
: Non-negative integer. Rolling feasibility window size.

progress
: Logical; if TRUE, emit progress information.

Value

A list with feasibility, selected design, operating characteristics, and full search results.

---

optimal_twostage_2arm_bf

Optimal two-stage two-arm Bayes-factor design for binary endpoints

---

Description

Computes an optimal two-stage two-arm Bayes-factor design for binary endpoints, minimizing the expected sample size under the null hypothesis while correcting the operating characteristics for the possibility of early stopping for futility.

Usage

```
optimal_twostage_2arm_bf(
  alpha = 0.05,
  beta = 0.2,
  k = 1/3,
  k_f = 3,
  n1_min = c(5, 5),
  n2_max = c(50, 50),
  alloc1 = 0.5,
  alloc2 = 0.5,
  power_cushion = 0,
  pceH0 = NULL,
  interim_fraction = c(0, 1),
  grid_step = 1L,
  coarse_step = 10L,
  progress = TRUE,
  max_iter = 10000L,
  ncores = getOption("bfbin2arm.ncores", 1L),
  compute_freq_oc = NULL,
  calibration_mode = c("Bayesian", "frequentist", "hybrid"),
  calibration_EN = NULL,
  p1_EN_H0 = NULL,
  p2_EN_H0 = NULL,
  alpha_freq = alpha,
  beta_freq = beta,
```

```
  p1_power = NULL,
  p2_power = NULL,
  p_null_grid = NULL,
  test = "BF01",
  a_0_d = 1,
  b_0_d = 1,
  a_0_a = 1,
  b_0_a = 1,
  a_1_d = 1,
  b_1_d = 1,
  a_2_d = 1,
  b_2_d = 1,
  a_1_a = 1,
  b_1_a = 1,
  a_2_a = 1,
  b_2_a = 1,
  a_1_d_Hminus = 1,
  b_1_d_Hminus = 1,
  a_2_d_Hminus = 1,
  b_2_d_Hminus = 1,
  a_1_a_Hminus = 1,
  b_1_a_Hminus = 1,
  a_2_a_Hminus = 1,
  b_2_a_Hminus = 1
)
```

Arguments

alpha	Numeric scalar, Bayesian type-I-error target.
beta	Numeric scalar, 1 minus the minimal Bayesian power target.
k	Numeric scalar, efficacy threshold; evidence against the null hypothesis is declared when the corresponding Bayes factor is smaller than k.
k_f	Numeric scalar, futility threshold; compelling evidence for the null hypothesis is declared when the corresponding Bayes factor is at least k_f.
n1_min	Numeric vector of length 2, minimum interim sample sizes for arms 1 and 2.
n2_max	Numeric vector of length 2, maximum final sample sizes for arms 1 and 2.
alloc1, alloc2	Positive numbers, allocation probabilities to arms 1 and 2.
power_cushion	Numeric scalar, optional extra power cushion used in the fixed-sample search of step 1.
pceH0	Optional numeric scalar in [0,1]. If specified, candidate two-stage designs must satisfy corrected CE_H0 >= pceH0.
interim_fraction	Numeric vector of length 2 giving lower and upper bounds for the interim sample size in each arm as a fraction of the fixed sample size.
grid_step	Positive integer giving the spacing of the interim design grid.

`coarse_step` Positive integer giving the spacing of the coarse fixed-sample search grid in step 1.

`progress` Logical; if `TRUE`, prints progress information.

`max_iter` Integer, maximum number of total fixed-sample sizes searched in step 1.

`ncores` Integer; number of parallel worker processes to use in the calibration. Defaults to `getOption("bfbin2arm.ncores", 1L)`. In vignettes and examples, a conservative value (e.g. 1 or 2) is recommended for CRAN checks, whereas users can increase this to exploit all available cores on their own machines.

`compute_freq_oc` Logical or `NULL`. Controls whether frequentist operating characteristics are computed for candidate two-stage designs during the search.

`calibration_mode` Character string specifying the calibration mode. Must be one of `"Bayesian"`, `"frequentist"`, or `"hybrid"`.

`calibration_EN` Character string or `NULL` specifying whether the design is ranked by Bayesian or frequentist expected sample size under the null hypothesis.

`p1_EN_H0`, `p2_EN_H0` Numeric scalars specifying the null response probabilities in control and treatment arm used when `calibration_EN = "frequentist"`.

`alpha_freq` Numeric scalar, frequentist type-I error target.

`beta_freq` Numeric scalar, 1 minus the frequentist power target.

`p1_power`, `p2_power` Numeric scalars specifying the success probabilities in control and treatment arm used for the frequentist power calculation.

`p_null_grid` Optional numeric vector giving the grid of null response probabilities used for frequentist type-I-error maximization. If `NULL`, a default grid is used.

`test` Character string, one of `"BF01"`, `"BF+0"`, `"BF-0"`, `"BF+-"`.

`a_0_d`, `b_0_d`, `a_0_a`, `b_0_a` Shape parameters for design and analysis priors under H_0.

`a_1_d`, `b_1_d`, `a_2_d`, `b_2_d` Shape parameters for design priors under H_1 or H_+.

`a_1_a`, `b_1_a`, `a_2_a`, `b_2_a` Shape parameters for analysis priors under H_1 or H_+.

`a_1_d_Hminus`, `b_1_d_Hminus`, `a_2_d_Hminus`, `b_2_d_Hminus` Optional design priors under H_- for directional tests.

`a_1_a_Hminus`, `b_1_a_Hminus` Shape parameters of the analysis prior under the directional null hypothesis H0- for arm 1.

`a_2_a_Hminus`, `b_2_a_Hminus` Shape parameters of the analysis prior under the directional null hypothesis H0- for arm 2.

Value

A list with the following components:

design	Four-element integer vector containing the selected two-stage design: interim sample sizes in arms 1 and 2 followed by final sample sizes in arms 1 and 2.
naive_oc	Named list of uncorrected fixed-sample operating characteristics and fixed-sample sizes found in step 1.
occ	Named numeric vector of corrected Bayesian operating characteristics for the selected two-stage design.
priors	List storing design hyperparameters and search settings.
freq_occ	Named numeric vector with fixed-sample and two-stage frequentist operating characteristics for the final design when frequentist calibration or reporting is active; otherwise NULL.
conv	Character string describing the search outcome. Typical values include "converged", "no_feasible_fixed", "no_interim_grid", and "no_feasible_design". In frequentist or hybrid calibration modes, additional informative status values may be returned when the best available design is returned although all requested constraints were not fully satisfied.

Examples

```
## Fast Bayesian example with small search space
res <- optimal_twostage_2arm_bf(
  alpha = 0.10,
  beta = 0.20,
  k = 1 / 3,
  k_f = 3,
  n1_min = c(3, 3),
  n2_max = c(12, 12),
  alloc1 = 0.5,
  alloc2 = 0.5,
  power_cushion = 0,
  pceH0 = NULL,
  interim_fraction = c(0.25, 0.75),
  grid_step = 2L,
  coarse_step = 4L,
  progress = FALSE,
  max_iter = 24L,
  calibration_mode = "Bayesian",
  test = "BF01",
  a_0_d = 1, b_0_d = 1,
  a_0_a = 1, b_0_a = 1,
  a_1_d = 1, b_1_d = 1,
  a_2_d = 1, b_2_d = 1,
  a_1_a = 1, b_1_a = 1,
  a_2_a = 1, b_2_a = 1
)
res$design
res$occ
```

```
res2 <- optimal_twostage_2arm_bf(
  alpha = 0.05,
  beta = 0.20,
  k = 1 / 3,
  k_f = 3,
  n1_min = c(5, 5),
  n2_max = c(20, 20),
  alloc1 = 0.5,
  alloc2 = 0.5,
  power_cushion = 0.02,
  pceH0 = 0.50,
  interim_fraction = c(0.25, 0.75),
  grid_step = 1L,
  coarse_step = 4L,
  progress = FALSE,
  max_iter = 40L,
  calibration_mode = "Bayesian",
  test = "BF+0",
  a_0_d = 1, b_0_d = 1,
  a_0_a = 1, b_0_a = 1,
  a_1_d = 1, b_1_d = 2,
  a_2_d = 2, b_2_d = 1,
  a_1_a = 1, b_1_a = 1,
  a_2_a = 1, b_2_a = 1,
  a_1_d_Hminus = 1, b_1_d_Hminus = 1,
  a_2_d_Hminus = 1, b_2_d_Hminus = 1,
  a_1_a_Hminus = 1, b_1_a_Hminus = 1,
  a_2_a_Hminus = 1, b_2_a_Hminus = 1
)
res2$design
res2$occ
```

`optimal_twostage_singlearm_bf` *Optimal two-stage single-arm Bayes factor design*

Description

Searches over admissible two-stage single-arm designs with a binary endpoint and returns the feasible design with smallest expected sample size under `H0`.

Usage

```
optimal_twostage_singlearm_bf(
  n1_min,
  n2_max,
```

```
  k,
  k_f,
  p0,
  a0 = 1,
  b0 = 1,
  a1 = 1,
  b1 = 1,
  dp = NA_real_,
  da0 = 1,
  db0 = 1,
  da1 = 1,
  db1 = 1,
  type = c("point", "direction"),
  calibration = c("Bayesian", "frequentist", "hybrid", "full"),
  target_power = 0.8,
  target_type1 = 0.05,
  target_ce_h0 = 0,
  target_freq_power = 0.8,
  target_freq_type1 = 0.05,
  power_cushion = 0
)
```

Arguments

n1_min	Minimum admissible interim sample size.
n2_max	Maximum admissible final sample size.
k	Efficacy threshold on the BF01 scale.
k_f	Futility threshold on the BF01 scale.
p0	Null response probability.
a0, b0	Beta analysis-prior parameters under H0.
a1, b1	Beta analysis-prior parameters under H1.
dp	Optional fixed point alternative used for frequentist power.
da0, db0	Beta design-prior parameters under H0.
da1, db1	Beta design-prior parameters under H1.
type	Character string; one of "point" or "direction".
calibration	Character string; one of "Bayesian", "frequentist", "hybrid", or "full".
target_power	Target corrected Bayesian power.
target_type1	Target corrected Bayesian type-I error.
target_ce_h0	Optional lower bound on corrected Bayesian compelling evidence in favour of H0.
target_freq_power	Target corrected frequentist power at dp.
target_freq_type1	Target corrected frequentist type-I error at p0.
power_cushion	Optional additive cushion for the fixed-sample power target in the first step of the search.

Details

Analysis priors are specified separately under `H0` and `H1` via `a0, b0, a1, b1`. Design priors are specified separately under `H0` and `H1` via `da0, db0, da1, db1`.

Value

A list describing the optimal design and search results.

`plot.bf_ttest_design` *Plot method for calibrated Bayesian t-test designs*

Description

Plot method for calibrated Bayesian t-test designs

Usage

```
## S3 method for class 'bf_ttest_design'
plot(
  x,
  which = c("decision", "analysis_prior", "design_prior", "n_curve"),
  annotate = TRUE,
  prior_points = 801L,
  ...
)
```

Arguments

`x`	Object of class `bfttestdesign`.
`which`	Character vector choosing panels among `"decision"`, `"analysisprior"`, `"designprior"`, `"effect"`, and `"ncurve"`.
`annotate`	Logical; add global title/subtitle.
`prior_points`	Number of grid points for prior density curves.
`...`	Ignored.

Value

A `ggplot` object or a patchwork object.

plot.bfbin2armrope2armdesign

Plot a two-arm one-stage ROPE equivalence design

Description

Plot a two-arm one-stage ROPE equivalence design

Usage

```
## S3 method for class 'bfbin2armrope2armdesign'
plot(
  x,
  type = c("overview", "operatingcharacteristics", "heatmap", "prior"),
  ...
)
```

Arguments

x	A bfbin2armrope2armdesign object.
type	Plot type.
...	Currently unused.

Value

Invisibly returns x.

plot.singlearm_bf_design

Plot a single-arm Bayes factor design

Description

Produces a diagnostic plot for a fitted single-arm two-stage Bayes factor design. Depending on the available information in the object, the plot shows the interim-search results, selected operating characteristics, and the design and analysis priors under H_0 and H_1.

Usage

```
## S3 method for class 'singlearm_bf_design'
plot(x, ...)
```

Arguments

x	An object of class "singlearm_bf_design".
...	Currently unused.

Value

The input object x, invisibly.

See Also

summary.singlearm_bf_design(), design_singlearm_bf(), optimal_twostage_singlearm_bf()

---

plot.singlearm_onestage_bf_design

Plot a one-stage single-arm BF design

Description

Produces a 2x2 figure: top-left: operating characteristic curves; top-right: table-like summary; bottom-left: design priors under H0 and H1; bottom-right: analysis priors under H0 and H1.

Usage

```
## S3 method for class 'singlearm_onestage_bf_design'
plot(
  x,
  what = c("all", "oc"),
  legend_pos = "right",
  legend_inset = 0,
  col_h0 = "#0072B2",
  col_h1 = "#D55E00",
  prior_lwd = 2,
  ...
)
```

Arguments

x	An object of class "singlearm_onestage_bf_design".
what	Character string; one of "all" or "oc".
legend_pos	Position passed to legend().
legend_inset	Numeric inset for legend().
col_h0	Colour for H0 priors in bottom panels.
col_h1	Colour for H1 priors in bottom panels.
prior_lwd	Line width for prior density curves.
...	Currently unused.

Value

Invisibly returns x.

`plot.singlearm_rope_twostage_design` *Plot a single-arm two-stage ROPE design*

Description

Produces a six-panel figure in a 2-row by 3-column layout:

Top-left Predictive type-I error and power as functions of the stage-1 sample size n_1 for the fixed maximum n^*, with the optimal n_1 marked. Optional dashed frequentist curves are added when corresponding calibration targets are present.

Top-centre Predictive PCE under H_0 as a function of n_1 for the fixed maximum n^*, with the optimal n_1 marked.

Top-right Compact textual summary of the optimal design and its operating characteristics.

Bottom-left Null design prior density with the direction-appropriate ROPE region shaded.

Bottom-centre Alternative design prior density with the direction-appropriate ROPE region shaded.

Bottom-right Analysis prior density with the direction-appropriate ROPE region shaded.

Usage

```
## S3 method for class 'singlearm_rope_twostage_design'
plot(x, type = c("default", "interim", "final"), ...)
```

Arguments

`x`	An object of class `"singlearm_rope_twostage_design"`.
`type`	Character string selecting the plot type. One of `"default"` (default), `"interim"`, or `"final"`.
`...`	Further graphical parameters passed to `par()` for the `"interim"` and `"final"` types.

Details

Additional plot types are available via the `type` argument:

`"default"` The 2x3 summary layout described above (default).

`"interim"` Interim posterior probability vs. stage-1 responses, showing the continuation region C_1.

`"final"` Final posterior probability vs. total responses, showing the acceptance region.

Value

Invisibly returns `x`.

plot.twoarm_rope_directional_onestage_design *Plot a two-arm directional ROPE design*

Description

Plot a two-arm directional ROPE design

Usage

```
## S3 method for class 'twoarm_rope_directional_onestage_design'
plot(
  x,
 which = c("default", "oc", "summary", "analysis_prior", "design_prior", "heatmap"),
  ...
)
```

Arguments

x	An object of class twoarm_rope_directional_onestage_design.
which	Character, one of "default" (six-panel layout), "oc", "summary", "analysis_prior", "design_prior", or "heatmap".
...	Currently unused.

plot_twostage_2arm_bf *Plot an optimal two-stage two-arm Bayes factor design*

Description

Given the result from optimal_twostage_2arm_bf(), this function produces a six-panel base R plot showing the design schematic, operating characteristics, and the design and analysis priors under H_0 and H_1.

Usage

```
plot_twostage_2arm_bf(
  res,
  main = "Optimal two-stage two-arm Bayes factor design"
)
```

Arguments

res	A list returned by optimal_twostage_2arm_bf(), containing components $design, $naive_oc, $occ and $priors.
main	Character string with the main title of the plot.

Value

Invisibly returns NULL; called for its side effect of producing a plot.

Examples

```
res <- optimal_twostage_2arm_bf(
  alpha = 0.10, beta = 0.20, k = 1/3, k_f = 3,
  n1_min = c(3, 3), n2_max = c(8, 8),
  alloc1 = 0.5, alloc2 = 0.5,
  power_cushion = 0,
  interim_fraction = c(0.5, 0.5),
  grid_step = 1,
  progress = FALSE,
  max_iter = 16,
  test = "BF01",
  a_0_d = 1, b_0_d = 1,
  a_0_a = 1, b_0_a = 1,
  a_1_d = 1, b_1_d = 1,
  a_2_d = 1, b_2_d = 1,
  a_1_a = 1, b_1_a = 1,
  a_2_a = 1, b_2_a = 1
)
if (is.numeric(res$design) && length(res$design) == 4 && !anyNA(res$design)) {
  plot_twostage_2arm_bf(res)
}
```

---

postProbHminus *Posterior probability P(p2 <= p1 | data)*

Description

Posterior probability P(p2 <= p1 | data)

Usage

```
postProbHminus(y1, y2, n1, n2, a_1_a = 1, b_1_a = 1, a_2_a = 1, b_2_a = 1)
```

Arguments

y1	Number of successes in arm 1 (control).
y2	Number of successes in arm 2 (treatment).
n1	Sample size in arm 1.
n2	Sample size in arm 2.
a_1_a, b_1_a	Shape parameters of the Beta prior for p_1 under the alternative (analysis prior).
a_2_a, b_2_a	Shape parameters of the Beta prior for p_2 under the alternative (analysis prior).

Value

Numeric scalar, posterior probability $P(p_2 <= p_1 | y_1, y_2)$.

postProbHplus — *Posterior probability P(p2 > p1 | data) under independent Beta priors*

Description

Uses Beta posteriors induced by the analysis priors to compute $P(p_2 > p_1 \mid y_1, y_2)$.

Usage

```
postProbHplus(y1, y2, n1, n2, a_1_a = 1, b_1_a = 1, a_2_a = 1, b_2_a = 1)
```

Arguments

y1	Number of successes in arm 1 (control).
y2	Number of successes in arm 2 (treatment).
n1	Sample size in arm 1.
n2	Sample size in arm 2.
a_1_a, b_1_a	Shape parameters of the Beta prior for p_1 under the alternative (analysis prior).
a_2_a, b_2_a	Shape parameters of the Beta prior for p_2 under the alternative (analysis prior).

Value

Numeric scalar, posterior probability $P(p_2 > p_1 | y_1, y_2)$.

powerbinbf01_fixed — *Bayesian and frequentist operating characteristics for a fixed-sample single-arm BF design*

Description

Computes operating characteristics for a genuine fixed-sample single-arm binomial design with final efficacy decision based on BF01 <= k.

Usage

```
powerbinbf01_fixed(
  n,
  k,
  p0,
  a0 = 1,
  b0 = 1,
  a1 = 1,
  b1 = 1,
  da0 = 1,
```

```
  db0 = 1,
  da1 = 1,
  db1 = 1,
  dp = NA_real_,
  type = c("point", "direction"),
  k_ce = NULL,
  grid_size = 801L
)
```

Arguments

n	Integer scalar. Total sample size.
k	Numeric scalar. Efficacy threshold on the BF01 scale.
p0	Numeric scalar in (0, 1). Null response probability.
a0, b0	Numeric scalars. Beta analysis-prior parameters under H0.
a1, b1	Numeric scalars. Beta analysis-prior parameters under H1.
da0, db0	Numeric scalars. Beta design-prior parameters under H0.
da1, db1	Numeric scalars. Beta design-prior parameters under H1.
dp	Optional numeric scalar in (0,1). If supplied, frequentist power under H1 is computed at p = dp.
type	Character string. One of \"point\" or \"direction\".
k_ce	Optional numeric scalar greater than 1. Threshold for compelling evidence in favour of H0 on the BF01 scale.
grid_size	Integer number of grid points used for numerical averaging.

Details

Bayesian operating characteristics are computed under separate design priors:

- for type = "direction", Bayesian power averages over p > p0 under the H1 design prior truncated to (p0, 1], Bayesian type-I error averages over p <= p0 under the H0 design prior truncated to [0, p0], and CE(H0) is averaged over the same truncated H0 design prior;
- for type = "point", Bayesian power averages under the H1 design prior on (0, 1), Bayesian type-I error is evaluated at the point null p = p0, and CE(H0) is also evaluated at p = p0.

Value

A list with Bayesian and frequentist operating characteristics for the fixed-sample design.

`powerbinbf01seq` *Bayesian and frequentist operating characteristics for a single-arm two-stage BF design*

Description

Computes naive fixed-sample and corrected two-stage operating characteristics for a single-arm binomial design with one interim analysis for futility. The Bayes factor is oriented as `BF01`, so efficacy corresponds to small values (`BF01 <= k`) and futility corresponds to large values (`BF01 >= kf`).

Usage

```
powerbinbf01seq(
  n1,
  n2,
  k,
  kf,
  p0,
  a0 = 1,
  b0 = 1,
  a1 = 1,
  b1 = 1,
  da0 = 1,
  db0 = 1,
  da1 = 1,
  db1 = 1,
  dp = NA_real_,
  type = c("point", "direction"),
  k_ce = NULL,
  grid_size = 801L
)
```

Arguments

`n1`	Integer scalar. Interim sample size.
`n2`	Integer scalar. Final sample size, with `n1 < n2`.
`k`	Numeric scalar. Efficacy threshold on the `BF01` scale.
`kf`	Numeric scalar. Futility threshold on the `BF01` scale.
`p0`	Numeric scalar in `(0, 1)`. Null response probability.
`a0`, `b0`	Numeric scalars. Beta analysis-prior parameters under H0.
`a1`, `b1`	Numeric scalars. Beta analysis-prior parameters under H1.
`da0`, `db0`	Numeric scalars. Beta design-prior parameters under H0.
`da1`, `db1`	Numeric scalars. Beta design-prior parameters under H1.

dp	Optional numeric scalar in (0,1). If supplied, frequentist power under H1 is computed at p = dp.
type	Character string. One of \"point\" or \"direction\".
k_ce	Optional numeric scalar greater than 1. Threshold for compelling evidence in favour of H0 on the BF01 scale.
grid_size	Integer number of grid points used for numerical averaging.

Details

Bayesian operating characteristics are computed under separate design priors:

- for type = "direction", Bayesian power averages over p > p0 under the H1 design prior truncated to (p0, 1], and Bayesian type-I error averages over p <= p0 under the H0 design prior truncated to [0, p0];
- for type = "point", Bayesian power averages under the H1 design prior on (0, 1), and Bayesian type-I error is evaluated at the point null p = p0.

If dp is supplied, additional frequentist power under H1 is computed at the fixed point alternative p = dp. Frequentist type-I error is computed at p = p0.

Value

A list with Bayesian and frequentist operating characteristics.

---

powertwoarmbinbf01 *Bayesian power, type-I error, and PCE(H0) for two-arm binomial Bayes factors*

Description

Computes Bayesian power, Bayesian type-I error, and the probability of compelling evidence under H_0 (or H_- for BF+-), for a given sample size and Bayes factor test. Optionally, frequentist type-I error and frequentist power are computed by summing over the rejection region.

Usage

```
powertwoarmbinbf01(
  n1,
  n2,
  k = 1/3,
  k_f = 1/3,
  test = c("BF01", "BF+0", "BF-0", "BF+-"),
  a_0_d = 1,
  b_0_d = 1,
  a_0_a = 1,
  b_0_a = 1,
  a_1_d = 1,
```

```
  b_1_d = 1,
  a_2_d = 1,
  b_2_d = 1,
  a_1_a = 1,
  b_1_a = 1,
  a_2_a = 1,
  b_2_a = 1,
 output = c("numeric", "predDensmatrix", "t1ematrix", "ceH0matrix", "frequentist_t1e"),
  a_1_d_Hminus = 1,
  b_1_d_Hminus = 1,
  a_2_d_Hminus = 1,
  b_2_d_Hminus = 1,
  compute_freq_t1e = FALSE,
  p1_grid = seq(0.01, 0.99, 0.02),
  p2_grid = seq(0.01, 0.99, 0.02),
  p1_power = NULL,
  p2_power = NULL,
  a_1_a_Hminus = 1,
  b_1_a_Hminus = 1,
  a_2_a_Hminus = 1,
  b_2_a_Hminus = 1
)
```

Arguments

n1, n2	Sample sizes in arms 1 and 2.
k	Evidence threshold for rejecting the null (inverted BF).
k_f	Evidence threshold for "compelling evidence" in favour of the null.
test	Character string, one of "BF01", "BF+0", "BF-0", "BF+-".
a_0_d, b_0_d, a_0_a, b_0_a	Shape parameters for design and analysis priors under H_0.
a_1_d, b_1_d, a_2_d, b_2_d	Shape parameters for design priors under H_1 or H_+.
a_1_a, b_1_a, a_2_a, b_2_a	Shape parameters for analysis priors under H_1 or H_+.
output	One of "numeric", "predDensmatrix", "t1ematrix", "ceH0matrix", "frequentist_t1e".
a_1_d_Hminus, b_1_d_Hminus, a_2_d_Hminus, b_2_d_Hminus	Optional design priors under H_- for directional tests.
compute_freq_t1e	Logical; if TRUE, compute frequentist type-I error over a grid.
p1_grid, p2_grid	Grids of true proportions for frequentist T1E.
p1_power, p2_power	Optional true proportions for frequentist power.
a_1_a_Hminus, b_1_a_Hminus, a_2_a_Hminus, b_2_a_Hminus	Shape parameters for analysis priors under H_- (directional tests).

Value

Depending on output, either a named numeric vector with components Power, Type1_Error, CE_H0 (and optionally frequentist metrics) or matrices of predictive densities.

Examples

```
# Basic Bayesian power for BF01 test
powertwoarmbinbf01(n1 = 30, n2 = 30, k = 1/3, test = "BF01")

# Directional test BF+0 with frequentist type-I error
powertwoarmbinbf01(n1 = 40, n2 = 40, k = 1/3, k_f = 3,
                   test = "BF+0", compute_freq_t1e = TRUE)

# Predictive density matrices (advanced)
powertwoarmbinbf01(n1 = 25, n2 = 25, output = "predDensmatrix")
```

---

predictiveDensityH0 *Predictive density under H0: p1 = p2 = p*

Description

Beta-binomial predictive density for data (y1,y2) under H0.

Usage

```
predictiveDensityH0(y1, y2, n1, n2, a_0_d = 1, b_0_d = 1)
```

Arguments

y1	Number of successes in arm 1 (control).
y2	Number of successes in arm 2 (treatment).
n1	Sample size in arm 1.
n2	Sample size in arm 2.
a_0_d, b_0_d	Design-prior parameters for common p under H0.

Value

Numeric scalar, predictive density.

predictiveDensityH1 *Predictive density under H1: p1 != p2*

Description

Product of two independent Beta-binomial predictive densities.

Usage

```
predictiveDensityH1(y1, y2, n1, n2, a_1_d = 1, b_1_d = 1, a_2_d = 1, b_2_d = 1)
```

Arguments

y1	Number of successes in arm 1 (control).
y2	Number of successes in arm 2 (treatment).
n1	Sample size in arm 1.
n2	Sample size in arm 2.
a_1_d, b_1_d	Design-prior parameters for p1.
a_2_d, b_2_d	Design-prior parameters for p2.

Value

Numeric scalar, predictive density.

predictiveDensityHminus_trunc *Predictive density under H-: p2 <= p1 (truncated prior)*

Description

Predictive density under H-: p2 <= p1 (truncated prior)

Usage

```
predictiveDensityHminus_trunc(
  y1,
  y2,
  n1,
  n2,
  a_1_d = 1,
  b_1_d = 1,
  a_2_d = 1,
  b_2_d = 1
)
```

Arguments

y1	Number of successes in arm 1 (control).
y2	Number of successes in arm 2 (treatment).
n1	Sample size in arm 1.
n2	Sample size in arm 2.
a_1_d, b_1_d	Design-prior parameters for p1.
a_2_d, b_2_d	Design-prior parameters for p2.

Value

Numeric scalar, predictive density under H-.

---

predictiveDensityHplus_trunc

Predictive density under H+: p2 > p1 (truncated prior)

Description

Predictive density under H+: p2 > p1 (truncated prior)

Usage

```
predictiveDensityHplus_trunc(
  y1,
  y2,
  n1,
  n2,
  a_1_d = 1,
  b_1_d = 1,
  a_2_d = 1,
  b_2_d = 1
)
```

Arguments

y1	Number of successes in arm 1 (control).
y2	Number of successes in arm 2 (treatment).
n1	Sample size in arm 1.
n2	Sample size in arm 2.
a_1_d, b_1_d	Design-prior parameters for p1.
a_2_d, b_2_d	Design-prior parameters for p2.

Value

Numeric scalar, predictive density under H+.

print.singlearm_onestage_bf_design

Print method for one-stage single-arm BF designs

Description

Print method for one-stage single-arm BF designs

Usage

```
## S3 method for class 'singlearm_onestage_bf_design'
print(x, ...)
```

Arguments

x	An object of class "singlearm_onestage_bf_design".
...	Currently unused.

Value

The input object x, invisibly.

print.singlearm_rope_twostage_design

Print a single-arm two-stage ROPE design

Description

Print a single-arm two-stage ROPE design

Usage

```
## S3 method for class 'singlearm_rope_twostage_design'
print(x, ...)
```

Arguments

x	An object of class "singlearm_rope_twostage_design".
...	Further arguments (currently ignored).

Value

Invisibly returns x.

print.summary.singlearm_bf_design

Print method for summary.singlearm_bf_design

Description

Print method for summary.singlearm_bf_design

Usage

```
## S3 method for class 'summary.singlearm_bf_design'
print(x, ...)
```

Arguments

x	An object of class "summary.singlearm_bf_design".
...	Additional arguments passed to or from other methods.

Value

The input object x, invisibly.

print.summary.singlearm_onestage_bf_design

Print method for summaries of one-stage single-arm BF designs

Description

Print method for summaries of one-stage single-arm BF designs

Usage

```
## S3 method for class 'summary.singlearm_onestage_bf_design'
print(x, digits = 3, ...)
```

Arguments

x	An object of class "summary.singlearm_onestage_bf_design".
digits	Number of digits to print.
...	Currently unused.

Value

The input object x, invisibly.

print.summary.singlearm_rope_twostage_design *Print a compact summary of a single-arm two-stage ROPE design*

Description

Print a compact summary of a single-arm two-stage ROPE design

Usage

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

Arguments

x	An object of class "summary.singlearm_rope_twostage_design".
digits	Number of digits used to display operating characteristics.
...	Further arguments (currently ignored).

priorProbHminus *Prior probability P(p2 <= p1) under independent Beta priors*

Description

Prior probability P(p2 <= p1) under independent Beta priors

Usage

```
priorProbHminus(a_1_a, b_1_a, a_2_a, b_2_a)
```

Arguments

a_1_a, b_1_a	Shape parameters of the Beta prior for p_1.
a_2_a, b_2_a	Shape parameters of the Beta prior for p_2.

Value

Numeric scalar, prior probability P(p2 <= p1).

priorProbHplus *Prior probability P(p2 > p1) under independent Beta priors*

Description

Prior probability P(p2 > p1) under independent Beta priors

Usage

```
priorProbHplus(a_1_a, b_1_a, a_2_a, b_2_a)
```

Arguments

a_1_a, b_1_a	Shape parameters of the Beta prior for p_1.
a_2_a, b_2_a	Shape parameters of the Beta prior for p_2.

Value

Numeric scalar, prior probability P(p2 > p1).

summary.singlearm_bf_design *Summary for single-arm BF designs*

Description

Summary for single-arm BF designs

Usage

```
## S3 method for class 'singlearm_bf_design'
summary(object, ...)
```

Arguments

object	An object of class "singlearm_bf_design".
...	Additional arguments (currently unused).

Value

An object of class "summary.singlearm_bf_design".

summary.singlearm_onestage_bf_design
Summarize a one-stage single-arm BF design

Description

Summarize a one-stage single-arm BF design

Usage

```
## S3 method for class 'singlearm_onestage_bf_design'
summary(object, ...)
```

Arguments

object	An object of class "singlearm_onestage_bf_design".
...	Currently unused.

Value

An object of class "summary.singlearm_onestage_bf_design".

summary.singlearm_rope_twostage_design
Summary for a single-arm two-stage ROPE design

Description

Summary for a single-arm two-stage ROPE design

Usage

```
## S3 method for class 'singlearm_rope_twostage_design'
summary(object, ...)
```

Arguments

object	An object of class "singlearm_rope_twostage_design".
...	Further arguments (currently ignored).

Value

Invisibly returns object.

`twoarmbinbf_plus0_direct` *Bayes factor BF+0 for the directional alternative vs point-null*

Description

Computes the Bayes factor BF_{+0} comparing the directional alternative hypothesis H_+ (p_2 > p_1) against the point-null H_0 (p_1 = p_2).

Usage

```
twoarmbinbf_plus0_direct(
  y1,
  y2,
  n1,
  n2,
  a_0_a = 1,
  b_0_a = 1,
  a_1_a = 1,
  b_1_a = 1,
  a_2_a = 1,
  b_2_a = 1
)
```

Arguments

`y1`, `y2`	Integer counts of successes in arms 1 and 2.
`n1`, `n2`	Integer sample sizes in arms 1 and 2.
`a_0_a`, `b_0_a`	Shape parameters of the **analysis** prior for the common response probability under H_0.
`a_1_a`, `b_1_a`	Shape parameters of the **analysis** prior for the response probability in arm 1 under H_+.
`a_2_a`, `b_2_a`	Shape parameters of the **analysis** prior for the response probability in arm 2 under H_+.

Details

Both marginal likelihoods are formed using the **analysis** priors, which represent inferential beliefs at the time the data are evaluated. The design priors are used only for computing Bayesian operating characteristics (predictive power / type-I error) and play no role here.

Value

Numeric scalar; the Bayes factor $BF_{+0} = m_+(y_1, y_2)/m_0(y_1, y_2)$.

`twoarmbinbf01` *Two-arm binomial Bayes factor BF01*

Description

Computes the Bayes factor BF_{01} comparing the point-null $H_0 : p_1 = p_2$ to the alternative $H_1 : p_1 \neq p_2$ in a two-arm binomial setting with Beta priors.

Usage

```
twoarmbinbf01(
  y1,
  y2,
  n1,
  n2,
  a_0_a = 1,
  b_0_a = 1,
  a_1_a = 1,
  b_1_a = 1,
  a_2_a = 1,
  b_2_a = 1
)
```

Arguments

`y1`	Number of successes in arm 1 (control).
`y2`	Number of successes in arm 2 (treatment).
`n1`	Sample size in arm 1.
`n2`	Sample size in arm 2.
`a_0_a`, `b_0_a`	Shape parameters of the Beta prior for the common-p under the null model (analysis prior).
`a_1_a`, `b_1_a`	Shape parameters of the Beta prior for p_1 under the alternative (analysis prior).
`a_2_a`, `b_2_a`	Shape parameters of the Beta prior for p_2 under the alternative (analysis prior).

Value

Numeric scalar, the Bayes factor BF_{01}.

Examples

```
twoarmbinbf01(10, 20, 30, 30)
```

Index

.sustained_singlearm_feasibility, 3

as.data.frame.singlearm_onestage_bf_design, 4
as.data.frame.singlearm_rope_twostage_design, 5

BFminus0, 5
BFminus1, 6
BFplus0, 6
BFplus1, 7
BFplusMinus, 7

calibrate_bf_ttest, 8
check_feasibility_rope, 10
coef.singlearm_rope_twostage_design, 11

design_singlearm_bf, 12
design_singlearm_bf(), *52*
design_singlearm_onestage_bf, 14
design_singlearm_onestage_rope, *10*, 16
design_singlearm_twostage_rope, 17
design_singlearm_twostage_rope(), *19–21*
design_singlearm_twostage_rope_equiv, 19
design_singlearm_twostage_rope_ni, 20
design_singlearm_twostage_rope_sup, 20
design_twoarm_onestage_bf, 21
design_twoarm_onestage_rope, 23
design_twoarm_onestage_rope_directional, 26
design_twoarm_onestage_rope_directional(), *28–30*
design_twoarm_onestage_rope_ni, 28
design_twoarm_onestage_rope_nisup, 29
design_twoarm_onestage_rope_sup, 30
design_twoarm_twostage_bf, 30

make_analysis_prior_greater, 33
make_analysis_prior_less_equal, 34
make_analysis_prior_two_sided, 34
make_design_prior_normal, 35
make_design_prior_point, 35
make_point_prior, 36
make_prior, 36
make_prior(), *35*

ntwoarmbinbf01, 37

optimal_onestage_singlearm_bf, 40
optimal_onestage_twoarm_bf, 42
optimal_twostage_2arm_bf, 44
optimal_twostage_singlearm_bf, 48
optimal_twostage_singlearm_bf(), *52*

plot.bf_ttest_design, 50
plot.bfbin2armrope2armdesign, 51
plot.singlearm_bf_design, 51
plot.singlearm_onestage_bf_design, 52
plot.singlearm_rope_twostage_design, 53
plot.twoarm_rope_directional_onestage_design, 54
plot_twostage_2arm_bf, 54
postProbHminus, 55
postProbHplus, 56
powerbinbf01_fixed, 56
powerbinbf01seq, 58
powertwoarmbinbf01, 59
predictiveDensityH0, 61
predictiveDensityH1, 62
predictiveDensityHminus_trunc, 62
predictiveDensityHplus_trunc, 63
print.singlearm_onestage_bf_design, 64
print.singlearm_rope_twostage_design, 64
print.summary.singlearm_bf_design, 65
print.summary.singlearm_onestage_bf_design, 65

print.summary.singlearm_rope_twostage_design, 66
priorProbHminus, 66
priorProbHplus, 67

summary.singlearm_bf_design, 67
summary.singlearm_bf_design(), *52*
summary.singlearm_onestage_bf_design, 68
summary.singlearm_rope_twostage_design, 68

twoarmbinbf01, 70
twoarmbinbf_plus0_direct, 69