Package ‘WCRBayesDesign’

April 10, 2026

Type Package

Title Bayesian Two-Stage Design with Window-Cohort and Controlled Roll-on for Time-to-Event Estimand

Version 1.0.1

Description Calibrates Bayesian two-stage designs for single-arm phase II trials with time-to-event endpoints using a window-cohort with controlled roll-on. Interim monitoring is anchored to a locked interim cohort and a pre-specified follow-up requirement, so analysis timing remains predictable while preserving follow-up maturity. The package searches feasible interim rules, optimizes final sample size and decision thresholds, evaluates operating characteristics by Monte Carlo simulation, and supports exponential, Weibull, log-normal, log-logistic, and user-defined baseline survival models. Related published foundations include Simon (1989) <doi:10.1016/0197-2456(89)90015-9> and Cotterill and Whitehead (2015) <doi:10.1002/sim.6426>.

License GPL-3

Encoding UTF-8

Depends R (>= 4.4.0)

Imports foreach, doParallel, stats, parallel

Suggests devtools, roxygen2, knitr, rmarkdown, testthat (>= 3.0.0)

RoxygenNote 7.3.3

NeedsCompilation no

Author Zhongheng Cai [aut, cre],
Haitao Pan [aut]

Maintainer Zhongheng Cai <`zhonghengcai123@gmail.com`>

Repository CRAN

Date/Publication 2026-04-10 10:00:08 UTC

Contents

`conduct` *Bayesian Interim and Final Analysis (Unified Transformed Time Framework)*

Description

Performs Bayesian analysis using the transformed proportional hazards model. By default, observed follow-up times are assumed to be on the original time scale and are transformed internally through the reference survival function S_0. Transformed times can also be supplied directly.

Usage

```
conduct(
  data,
  current.n,
  nmax,
  tau,
  theta_L,
  a0,
  b0,
  pIA,
  pF,
  S0_dist = c("exp", "weibull", "lognormal", "loglogistic", "custom"),
  S0_par = list(lambda = 0.05),
  S0_fun = NULL,
  time_scale = c("raw", "transformed")
)
```

Arguments

`data`	A `data.frame` with columns: • `Time`: Observed follow-up time, interpreted according to `time_scale`. • `Status`: Event indicator (1 = event, 0 = censored).
`current.n`	Integer. Current number of patients in the analysis.
`nmax`	Integer. Maximum total sample size.
`tau`	Numeric. Time horizon used to evaluate survival on the original scale.

`theta_L`	Numeric. Null-hypothesis survival threshold evaluated at `tau`.
`a0`, `b0`	Numeric. Hyperparameters for the Gamma prior.
`pIA`, `pF`	Numeric. Posterior probability thresholds for interim and final analyses.
`S0_dist`	Character. Baseline survival distribution type.
`S0_par`	List. Parameters of the baseline survival function.
`S0_fun`	Function. Baseline survival function (for custom).
`time_scale`	Character. Time scale supplied in `data$Time`: `"raw"` (default, original follow-up time) or `"transformed"`.

Value

A named list with the following components:

Current.sample.size Integer. The analysis sample size supplied via `current.n`.

Posterior.shape Numeric. Posterior Gamma shape parameter $a_0 + \sum \Delta_i$.

Posterior.rate Numeric. Posterior Gamma rate parameter $b_0 + \sum W_i$, where $W_i = -\log S_0(X_i)$ when `time_scale = "raw"`.

Posterior tail probability (Pr >= threshold) Numeric. Posterior probability $\Pr(\delta \geq \delta_{\text{goal}} \mid \text{data})$ used for the Go/No-Go or efficacy decision.

Posterior mean of survival rate Numeric. Posterior mean of the survival probability at τ, mapped back to the original survival scale via $S_0(\tau)^{\delta}$.

95% credible interval of survival rate Numeric vector of length 2. Lower and upper bounds of the posterior credible interval for the survival probability at τ.

Decision Character. Interim decision (`"Go (Continue)"` or `"No-Go (Stop for futility)"`) when `current.n < nmax`, or final decision (`"Efficacious"` or `"Not Efficacious"`) when `current.n >= nmax`.

Examples

```
dat <- data.frame(Time = c(5, 10, 15, 8, 20), Status = c(1, 0, 1, 1, 0))
conduct(dat, current.n = 5, nmax = 30, tau = 24, theta_L = 0.62,
        a0 = 0.01, b0 = 0.01, pIA = 0.5, pF = 0.05,
        S0_dist = "weibull", S0_par = list(k = 1.2, lambda = 0.08))
```

`delta_from_theta_goal` *Convert target EFS goal to delta parameter under proportional hazards model*

Description

Converts a target event-free survival (EFS) probability at a given time τ to the corresponding δ_{goal} parameter under the proportional hazards assumption, where $S(t) = S_0(t)^{\delta}$.

Usage

```
delta_from_theta_goal(theta_goal, tau, S0_fun, S0_par = NULL)
```

Arguments

`theta_goal`	Numeric. Target EFS probability at time τ.
`tau`	Numeric. Follow-up time at which the EFS probability is defined.
`S0_fun`	Function. Reference survival function returning values in (0, 1].
`S0_par`	List. Parameters passed to `S0_fun`.

Details

This function is used to translate a survival probability goal into the corresponding delta threshold under the transformed model framework.

Value

A numeric value giving the corresponding δ_{goal}.

Examples

```
# Example: Convert an EFS goal of 0.62 at tau = 24
delta_from_theta_goal(
  theta_goal = 0.62,
  tau = 24,
  S0_fun = S0_weibull,
  S0_par = list(k = 1.2, lambda = 0.08)
)
```

`find_Nw_pIA` *Search for feasible interim analysis thresholds (Unified Framework)*

Description

Performs grid search over interim sample sizes (N_w) and time offsets (X), identifying feasible posterior probability thresholds (p_IA) for early futility stopping. Uses unified transformed time framework.

Usage

```
find_Nw_pIA(
  tau,
  theta_L,
  theta_alt,
  alpha_target,
  beta_target,
```

```
  rate,
  X_grid,
  num_grid = 20,
  p_H0 = 0.4,
  p_H1 = 0.1,
  nsim = 10000,
  S0_dist = c("exp", "weibull", "lognormal", "loglogistic", "custom"),
  a0 = 0.01,
  b0 = 0.01,
  S0_par = list(lambda = 0.05),
  S0_fun = NULL,
  pIA_grid = seq(0.01, 0.99, by = 0.01),
  seed = 1234,
  method = "landmark",
  drop_na = TRUE
)
```

Arguments

`tau`	Numeric. Maximum follow-up time.
`theta_L`	Numeric. Target EFS probability at tau under H0.
`theta_alt`	Numeric. EFS probability at tau under H1.
`alpha_target`, `beta_target`	Numeric. Target Type I/II error rates.
`rate`	Numeric. Accrual rate (patients per unit time).
`X_grid`	Numeric vector. Offset times after enrollment of N_w-th patient.
`num_grid`	Integer. Number of grid points for N_w search.
`p_H0`, `p_H1`	Numeric. Required early stopping probabilities.
`nsim`	Integer. Number of Monte Carlo simulations per (N_w, X).
`S0_dist`	Character. Baseline survival distribution type.
`a0`, `b0`	Numeric. Hyperparameters for Gamma prior.
`S0_par`	List. Parameters for S0 distribution.
`S0_fun`	Function. Custom S0 function (optional).
`pIA_grid`	Numeric vector. Grid of candidate pIA thresholds.
`seed`	Integer. Random seed.
`method`	Character. Analysis method controlling administrative censoring: `"hr"` (proportional hazards, relaxed follow-up to calendar horizon) or `"landmark"` (strict landmark censoring, each subject capped at τ). Default is `"landmark"`.
`drop_na`	Logical. Remove rows with no feasible pIA.

Value

A data frame with one row per candidate (N_w, X) pair and the following columns:

Nw Integer. Interim-stage enrollment target used to trigger the waiting window.

X Numeric. Additional waiting time after enrollment of the N_w-th subject before the interim analysis.

pIA Numeric. Smallest feasible futility threshold identified on `pIA_grid` for this (N_w, X) combination.

piF_H0 Numeric. Estimated probability under H_0 that the trial stops for futility at interim using the selected `pIA`.

piF_H1 Numeric. Estimated probability under H_1 that the trial stops for futility at interim using the selected `pIA`.

Rows with no feasible `pIA` are removed when `drop_na = TRUE`.

Examples

```
find_Nw_pIA(
  tau = 24, theta_L = 0.62, theta_alt = 0.80,
  alpha_target = 0.05, beta_target = 0.20,
  rate = 5/12, X_grid = c(1, 2),
  nsim = 500, S0_dist = "exp",
  seed = 123
)
```

`oc_two_stage` *Evaluate operating characteristics for two-stage adaptive survival designs (Unified Framework)*

Description

Simulates operating characteristics (OC) using unified transformed time framework. All internal calculations use transformed times. Output times are reported in original scale.

Time Metrics (all in original time scale):

- `E_time_IA_H0/H1`: Expected time to interim analysis (unconditional mean, always occurs).
- `E_time_FA_H0/H1`: Expected time to final analysis **conditional on reaching FA** (i.e., mean over trials that did not stop early at IA).
- `E_follow_H0`: Expected total follow-up time under H0 (in original time scale).

Usage

```
oc_two_stage(
  N,
  Nw,
  X,
  pIA,
  pF,
  rate,
```

```
  tau,
  theta_L,
  theta_alt,
  a0,
  b0,
  nsim = 2000,
  S0_dist = c("exp", "weibull", "lognormal", "loglogistic", "custom"),
  S0_par = list(lambda = 0.05),
  S0_fun = NULL,
  S0_inv_fun = NULL,
  seed = NULL,
  method = "landmark"
)
```

Arguments

`N`	Integer. Total sample size.
`Nw`	Integer. Number of patients enrolled before interim analysis.
`X`	Numeric. Additional time after enrollment of N_w-th subject before IA.
`pIA`	Numeric. Posterior probability threshold for futility at IA.
`pF`	Numeric. Posterior probability threshold for efficacy at FA.
`rate`	Numeric. Accrual rate (patients per unit time).
`tau`	Numeric. Maximum follow-up duration per subject.
`theta_L`	Numeric. Event-free survival probability at tau under H0.
`theta_alt`	Numeric. Event-free survival probability at tau under H1.
`a0`, `b0`	Numeric. Prior hyperparameters for Gamma posterior.
`nsim`	Integer. Number of Monte Carlo simulation replicates.
`S0_dist`	Character. Baseline survival distribution type.
`S0_par`	List. Parameters for S0 distribution.
`S0_fun`	Function. Custom S0 function (optional).
`S0_inv_fun`	Function. Inverse baseline survival function for `S0_dist = "custom"`. If omitted, `S0_par$inv_fun` is used.
`seed`	Integer or NULL. Random seed for reproducibility.
`method`	Character. Analysis method controlling administrative censoring: `"hr"` (proportional hazards, relaxed follow-up to calendar horizon) or `"landmark"` (strict landmark censoring, each subject capped at τ). Default is `"landmark"`.

Value

A named list with estimated operating characteristics:

alpha Numeric. Estimated Type I error under H_0.

power Numeric. Estimated power under H_1.

ESP_H0 Numeric. Early stopping probability at interim under H_0.

ESP_H1 Numeric. Early stopping probability at interim under H_1.

ESS_H0 Numeric. Expected total number of enrolled subjects under H_0, accounting for interim stopping.

E_event_H0 Numeric. Expected number of observed events under H_0.

E_event_H1 Numeric. Expected number of observed events under H_1.

E_follow_H0 Numeric. Expected total observed follow-up under H_0, on the original time scale.

max_n_total_H0 Numeric. Maximum realized sample size under H_0.

mean_n_between Numeric. Mean number of subjects enrolled between the N_w-th enrollment and the interim analysis under H_0.

median_n_between Numeric. Median number of subjects enrolled between the N_w-th enrollment and the interim analysis under H_0.

mean_fu_between Numeric. Mean observed follow-up, on the original time scale, among subjects enrolled between the N_w-th enrollment and the interim analysis under H_0.

median_fu_between Numeric. Median observed follow-up, on the original time scale, among subjects enrolled between the N_w-th enrollment and the interim analysis under H_0.

E_time_IA_H0 Numeric. Expected calendar time to the interim analysis under H_0.

E_time_FA_H0 Numeric or `NA`. Expected calendar time to the final analysis under H_0, conditional on reaching final analysis.

E_time_IA_H1 Numeric. Expected calendar time to the interim analysis under H_1.

E_time_FA_H1 Numeric or `NA`. Expected calendar time to the final analysis under H_1, conditional on reaching final analysis.

Examples

```
oc_two_stage(
  N = 40, Nw = 20, X = 2, pIA = 0.5, pF = 0.05,
  rate = 5/12, tau = 24, theta_L = 0.62, theta_alt = 0.80,
  a0 = 0.01, b0 = 0.01, nsim = 500,
  S0_dist = "exp", seed = 123
)
```

`run_two_stage_trial` *Run Two-Stage Bayesian Single-Arm Survival Trial Simulation*

Description

Simulates a two-stage Bayesian adaptive single-arm survival trial. Performs an interim analysis after the first-stage sample (N_w) and decides whether to continue to the final analysis (N_{max}) based on a Bayesian posterior decision rule.

Usage

```
run_two_stage_trial(
  seed = NULL,
  tau,
  theta_L,
  theta_true,
  a0 = 0.01,
  b0 = 0.01,
  rate,
  Nw,
  Nmax,
  X,
  pIA,
  pF,
  dist = c("exp", "weibull", "lognormal", "loglogistic"),
  dist_par = list(lambda = 0.05, k = 1.2),
  model = "exp",
  method = "landmark",
  verbose = TRUE
)
```

Arguments

`seed`	Integer. Random seed for reproducibility.
`tau`	Numeric. Time horizon (τ) for evaluating survival.
`theta_L`	Numeric. Null-hypothesis survival threshold at τ.
`theta_true`	Numeric. True underlying survival probability at τ.
`a0`, `b0`	Numeric. Hyperparameters for Gamma(a_0, b_0) prior.
`rate`	Numeric. Accrual rate parameter (exponential inter-arrival times).
`Nw`	Integer. Interim-stage sample size.
`Nmax`	Integer. Maximum (final) sample size.
`X`	Numeric. Additional follow-up time window for interim analysis.
`pIA`, `pF`	Numeric. Posterior probability thresholds for interim and final decisions.
`dist`	Character. Distribution for generating event times. One of: • `"exp"`: Exponential with rate λ. • `"weibull"`: Weibull with shape k and rate λ. • `"lognormal"`: Log-normal with meanlog μ, sdlog σ. • `"loglogistic"`: Log-logistic with shape k, rate λ.
`dist_par`	List. Parameters for the chosen event-time distribution.
`model`	Deprecated. Retained for backward compatibility and ignored. Analysis always uses the unified transformed-time framework.
`method`	Character. Analysis method controlling administrative censoring: `"hr"` (proportional hazards, relaxed follow-up) or `"landmark"` (strict landmark censoring, each subject capped at τ). Default is `"landmark"`.

- `method = "landmark"`: strict censoring.
  - Interim analysis: follow-up cannot exceed the design horizon τ.
  - Final analysis: each subject is administratively censored at τ from randomization (patient-level horizon).
- `method = "hr"`: relaxed censoring.
  - Interim analysis: the maximum follow-up available by the interim calendar time, without truncation at τ.
  - Final analysis: the trial ends at $t_{final} = a_{(N)} + \tau$, where $a_{(N)}$ is the accrual time of the last enrolled subject; each subject is then censored at $f_i = t_{final} - a_i$.

`verbose`	Logical. If `TRUE`, emit progress messages during simulation. Set to `FALSE` to suppress console output.

Details

The function simulates accrual and event times for all N_{max} patients, performs an interim analysis using the first N_w patients, and, if the posterior probability is below `pIA`, continues to final analysis. At the final stage, all patients (including interim ones) are followed up to τ and analyzed again using the same Bayesian decision rule.

The internal analysis relies on the `conduct()` function for posterior inference.

Value

A named list describing one simulated trial trajectory:

data_interim Data frame for the interim analysis cohort. Columns are `Time` (observed raw follow-up), `Status` (event indicator), and `Time_transformed` (reference transformed time for the same records).

data_final Data frame for the final analysis cohort, with the same columns as `data_interim`, or `NULL` if the trial stopped at interim.

result_interim Named list returned by `conduct()` for the interim analysis.

result_final Named list returned by `conduct()` for the final analysis, or `NULL` if the trial stopped at interim.

accrual_times Numeric vector of calendar enrollment times for all simulated subjects.

event_times Numeric vector of true event times generated under the chosen data-generating distribution.

method Character. Administrative censoring method actually used in the simulation (`"landmark"` or `"hr"`).

Examples

```
run_two_stage_trial(
  seed = 123,
  tau = 12,
  theta_L = 0.4,
  theta_true = 0.6,
  a0 = 0.01, b0 = 0.01,
```

```
  rate = 0.5,
  Nw = 20, Nmax = 40,
  X = 3,
  pIA = 0.2, pF = 0.05,
  dist = "weibull",
  dist_par = list(k = 1.2, lambda = 0.08),
  method = "landmark",
  verbose = FALSE
)
```

`S0_inverse` *Inverse of baseline survival function S_0*

Description

Computes the inverse of the baseline survival function S_0. Given a survival probability S, returns the time t such that S_0(t) = S.

Usage

```
S0_inverse(
  surv_prob,
  dist = c("exp", "weibull", "lognormal", "loglogistic", "custom"),
  par
)
```

Arguments

`surv_prob`	Numeric vector. Survival probability values (between 0 and 1).
`dist`	Character. Distribution type: "exp", "weibull", "lognormal", "loglogistic", or "custom".
`par`	List. Parameters for the distribution: • For "exp": list(lambda) • For "weibull": list(k, lambda) (shape and rate; $S_0(t) = \exp\{-(\lambda t)^k\}$) • For "lognormal": list(meanlog, sdlog) • For "loglogistic": list(k, lambda) (shape and rate) • For "custom": list(inv_fun) where inv_fun is a function(S) returning t

Value

Numeric vector of times corresponding to the survival probabilities.

Examples

```
# Exponential
S0_inverse(0.5, "exp", list(lambda = 0.1))

# Weibull
S0_inverse(c(0.8, 0.5), "weibull", list(k = 1.2, lambda = 0.08))
```

`S0_weibull` *Weibull reference survival function*

Description

Computes the Weibull reference survival function $S_0(t) = \exp\{-(\lambda t)^k\}$ for given shape and rate parameters.

Usage

```
S0_weibull(t, k, lambda)
```

Arguments

`t`	Numeric vector of time points.
`k`	Numeric. Shape parameter of the Weibull distribution.
`lambda`	Numeric. Rate parameter of the Weibull distribution (reciprocal of the conventional scale; larger values imply shorter survival).

Details

This function can be used as a baseline survival model within the transformed analysis framework. It is also directly accessible to users for defining custom reference survival functions.

Value

A numeric vector of survival probabilities $S_0(t)$.

Examples

```
S0_weibull(c(0, 5, 10), k = 1.2, lambda = 0.08)
```

`stats_transformed` *Compute transformed statistics for survival data*

Description

Computes the sufficient statistics (d, U) used in the transformed survival model framework, where:

- $d = \sum \Delta_i$ is the total number of observed events, and
- $U = \sum -\log(S_0(X_i))$ is the cumulative transformed time using the reference survival function $S_0(t)$.

Usage

```
stats_transformed(X, Delta, S0_fun, S0_par)
```

Arguments

`X`	Numeric vector. Observed follow-up times from enrollment.
`Delta`	Integer or logical vector. Event indicator (1 = event, 0 = censored).
`S0_fun`	Function. Baseline survival function that returns values in (0, 1].
`S0_par`	List. Parameters passed to `S0_fun`.

Details

This function allows users to derive model-specific posterior parameters for transformed analyses, given observed follow-up times and event indicators.

Value

A named list with components:

d Integer. Total number of observed events $d = \sum \Delta_i$.

U Numeric. Total transformed follow-up $U = \sum -\log\{S_0(X_i)\}$. This is the exposure term that enters the Gamma posterior update for δ.

Examples

```
# Example with Weibull baseline
times <- c(5, 8, 12)
events <- c(1, 0, 1)
stats_transformed(times, events, S0_fun = S0_weibull, S0_par = list(k = 1.2, lambda = 0.08))
```

`two_stage_optimize_design` *Optimize two-stage adaptive survival design*

Description

Searches for optimal two-stage adaptive survival designs based on simulated operating characteristics and user-defined optimality criteria.

Usage

```
two_stage_optimize_design(
  NwX_pIA_results,
  rate,
  tau,
  theta_L,
  theta_alt,
  a0,
  b0,
  alpha_target,
  beta_target,
  nsim = 2000,
  S0_dist = c("exp", "weibull", "lognormal", "loglogistic", "custom"),
  S0_par = list(lambda = 0.05),
  S0_fun = NULL,
  S0_inv_fun = NULL,
  tol = 0.01,
  max_iter = 25,
  pF_lo = 0.01,
  pF_hi = 0.99,
  ncores = max(1, parallel::detectCores() - 1),
  seed = 1234,
  optimize = c("followup", "ESS", "minimax"),
  method = "landmark",
  verbose = TRUE
)
```

Arguments

`NwX_pIA_results`	Data frame. Output from `find_Nw_pIA()`, containing columns `Nw`, `X`, and `pIA`.
`rate`	Numeric. Accrual rate (patients per unit time).
`tau`	Numeric. Maximum follow-up duration per subject.
`theta_L`	Numeric. Event-free survival probability at τ under H0.
`theta_alt`	Numeric. Event-free survival probability at τ under H1.
`a0`, `b0`	Numeric. Prior hyperparameters for the Gamma posterior.

`alpha_target`	Numeric. Target Type I error rate.
`beta_target`	Numeric. Target Type II error rate.
`nsim`	Integer. Number of Monte Carlo simulations per candidate design.
`S0_dist`	Character. Baseline survival distribution type.
`S0_par`	List. Parameters passed to `S0_fun`.
`S0_fun`	Function. Reference survival function (used when `method = "trans"`).
`S0_inv_fun`	Function. Inverse baseline survival function for `S0_dist = "custom"`. If omitted, `S0_par$inv_fun` is used.
`tol`	Numeric. Tolerance for the binary search over final posterior thresholds.
`max_iter`	Integer. Maximum number of iterations during binary search.
`pF_lo`, `pF_hi`	Numeric. Lower and upper bounds for searching final posterior threshold.
`ncores`	Integer. Number of CPU cores for parallel computation.
`seed`	Integer. Random seed for reproducibility.
`optimize`	Character. Criterion used to select the optimal design: **"followup"** Minimize expected total follow-up time under H0 (default). **"ESS"** Minimize expected sample size under H0. **"minimax"** Minimize maximum realized sample size under H0.
`method`	Character. Analysis method controlling administrative censoring: `"hr"` (proportional hazards, relaxed follow-up to calendar horizon) or `"landmark"` (strict landmark censoring, each subject capped at τ). Default is `"landmark"`. • `method = "landmark"`: strict censoring. – Interim analysis: follow-up cannot exceed the design horizon τ. – Final analysis: each subject is administratively censored at τ from randomization (patient-level horizon). • `method = "hr"`: relaxed censoring. – Interim analysis: the maximum follow-up available by the interim calendar time, without truncation at τ. – Final analysis: the trial ends at $t_{final} = a_{(N)} + \tau$, where $a_{(N)}$ is the accrual time of the last enrolled subject; each subject is then censored at $f_i = t_{final} - a_i$.
`verbose`	Logical. Whether to print progress messages.

Details

The function evaluates candidate combinations of interim sample size (N_w), offset time (X), and total sample size (N) using Monte Carlo simulation, and identifies the design that satisfies Type I / II error constraints while minimizing a specified optimization criterion.

Each candidate design is simulated using `oc_two_stage()`. For each combination of N_w, X, and N, the function performs a binary search over the final posterior threshold `pF` to satisfy:

$$\text{power} \geq 1 - \beta_{\text{target}}, \quad \alpha \leq \alpha_{\text{target}}.$$

The design with the smallest selected optimization metric under H0 is reported as optimal.

Seed consistency with `find_Nw_pIA()`: For each (N_w, X) combination, the seed passed to `oc_two_stage()` is derived as `seed + Nw * 997L + round(X * 1000) * 13L`, matching the sub-seed formula used in `find_Nw_pIA()`. This ensures that the early stopping probability (`ESP_H0`) reported here is comparable to the `piF_H0` from `find_Nw_pIA()` when the same `seed` is used. To reproduce the OC of the best design independently, call `oc_two_stage(..., seed = best$seed_used)`.

Value

A list with two components:

all Data frame of all evaluated candidate designs. Each row corresponds to one feasible combination of (N_w, X, p_{IA}, N, p_F) together with its simulated operating characteristics. Columns include design parameters (`Nw`, `X`, `pIA`, `N`, `pF`), reproducibility metadata (`seed_used`), error-rate summaries (`alpha`, `power`, `ESP_H0`, `ESP_H1`), resource summaries (`ESS_H0`, `E_event_H0`, `E_event_H1`, `E_follow_H0`, `max_n_total_H0`), interim-gap summaries (`mean_n_between`, `median_n_between`, `mean_fu_between`, `median_fu_between`), calendar-time summaries (`E_time_IA_H0`, `E_time_FA_H0`, `E_time_IA_H1`, `E_time_FA_H1`), and a logical feasibility flag `feasible`.

best One-row data frame giving the optimal feasible design under the requested `optimize` criterion. This row has the same columns as `all`. Pass `best$seed_used` back to `oc_two_stage()` to reproduce the reported operating characteristics exactly. Returns `NULL` if no feasible design is found.

Examples

```
res_pIA <- find_Nw_pIA(
  tau = 24, theta_L = 0.62, theta_alt = 0.80,
  alpha_target = 0.05, beta_target = 0.20,
  rate = 5/12, X_grid = 1,
  num_grid = 3, nsim = 100,
  pIA_grid = seq(0.1, 0.9, by = 0.2),
  S0_dist = "exp", seed = 123
)

opt <- two_stage_optimize_design(
  NwX_pIA_results = res_pIA,
  rate = 5/12, tau = 24,
  theta_L = 0.62, theta_alt = 0.80,
  a0 = 0.01, b0 = 0.01,
  alpha_target = 0.05, beta_target = 0.20,
  nsim = 100, S0_dist = "exp",
  optimize = "ESS",
  ncores = 1, seed = 123, verbose = FALSE
)
opt$best
```

Index