Package: pob2ef (via r-universe)

November 18, 2026

Title Partially Optimized Bayesian Phase II Efficacy and Futility Design

Version 0.1.0

Description Implements a partially optimized Bayesian single-arm phase II design for a binary efficacy endpoint with efficacy and futility stopping. The design searches candidate response-count boundaries and, subject to prespecified error constraints, selects the design that minimizes the grey-zone at the final look. The package creates decision tables and evaluates operating characteristics by exact trial-path enumeration or Monte Carlo simulation.

License MIT + file LICENSE

Encoding UTF-8

Suggests knitr, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author Feinan Lu [aut, cre]

Maintainer Feinan Lu <feinan.lu@hotmail.com>

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

Date/Publication 2026-11-17 12:10:02 UTC

RemoteUrl https://github.com/cran/pob2ef

RemoteRef HEAD

RemoteSha bb8925cf23a8effe3bef798528d1f7d65069977e

Contents

decision_table 2
plot_decision_boundaries 2

plot_operating_characteristics 3
select_design 4
simulate_design 5

Index **7**

`decision_table` *Create a response-count decision table*

Description

Returns posterior evidence, probability boundaries, and the futility, superiority, continue, or final grey-zone action for every possible response count at each planned analysis.

Usage

```
decision_table(design)
```

Arguments

`design`	A `pob2ef_design` object returned by `select_design()`.

Value

A data frame with one row per possible cumulative response count at each planned look.

Examples

```
design <- select_design(
  n_max = 20, looks = c(10, 20), p0 = 0.20, p1 = 0.40,
  alpha = 0.10, beta = 0.20,
  lambda1_grid = 0.50, lambda2_grid = 0.20, gamma_grid = 0, eta_grid = 0
)
decision_table(design)
```

`plot_decision_boundaries`
Plot decision boundaries

Description

Displays every response-count action at each planned analysis as a filled grid. Interim continuation and final grey-zone outcomes are combined as the grey-zone display category.

Usage

```
plot_decision_boundaries(design, ...)
```

Arguments

design	A pob2ef_design object returned by select_design().
...	Additional arguments passed to graphics::plot().

Value

Invisibly returns the response-count decision table used for the figure.

Examples

```
design <- select_design(
  n_max = 20, looks = c(10, 20), p0 = 0.20, p1 = 0.40,
  alpha = 0.10, beta = 0.20,
  lambda1_grid = 0.50, lambda2_grid = 0.20, gamma_grid = 0, eta_grid = 0
)
plot_decision_boundaries(design)
```

---

plot_operating_characteristics

Plot simulated operating characteristics

Description

Plot simulated operating characteristics

Usage

```
plot_operating_characteristics(
  simulation,
  metric = "superiority_probability",
  ...
)
```

Arguments

simulation	A pob2ef_simulation object returned by simulate_design().
metric	Operating characteristic to display.
...	Additional arguments passed to graphics::plot().

Value

Invisibly returns the simulation results used for the figure.

Examples

```
design <- select_design(
  n_max = 20, looks = c(10, 20), p0 = 0.20, p1 = 0.40,
  alpha = 0.10, beta = 0.20,
  lambda1_grid = 0.50, lambda2_grid = 0.20, gamma_grid = 0, eta_grid = 0
)
simulation <- simulate_design(
  design, true_rates = c(0.20, 0.30, 0.40), n_sim = 100, seed = 2026
)
plot_operating_characteristics(
  simulation,
  metric = "superiority_probability"
)
```

---

select_design *Select a partially optimized Bayesian phase II design*

Description

Searches a grid of boundary parameters and returns the feasible candidate that minimizes the final grey-zone probability at the desirable response rate subject to the prespecified error constraints.

Usage

```
select_design(
  n_max,
  looks,
  p0,
  p1,
  alpha,
  beta,
  prior = c(1, 1),
  lambda1_grid = seq(0.05, 0.5, by = 0.05),
  lambda2_grid = seq(0.05, 0.45, by = 0.05),
  gamma_grid = 0:3,
  eta_grid = 0:3
)
```

Arguments

n_max	Maximum evaluable sample size.
looks	Increasing planned analysis sample sizes ending at n_max.
p0	Unacceptable response-rate benchmark.
p1	Clinically desirable response-rate target, strictly greater than p0.
alpha	Maximum type I error, defined as superiority probability at p0.
beta	Maximum type II error, defined as futility probability at p1.

prior	Two positive beta-prior shape parameters.
lambda1_grid	Candidate values for the futility boundary level.
lambda2_grid	Candidate values for the superiority boundary level.
gamma_grid	Candidate values for the futility boundary shape.
eta_grid	Candidate values for the superiority boundary shape.

Value

An object of class pob2ef_design containing the selected parameters, operating characteristics, feasible candidates, and all evaluated candidates.

Examples

```
design <- select_design(
  n_max = 20,
  looks = c(10, 20),
  p0 = 0.20,
  p1 = 0.40,
  alpha = 0.10,
  beta = 0.20,
  lambda1_grid = 0.50,
  lambda2_grid = 0.20,
  gamma_grid = 0,
  eta_grid = 0
)
design$selected
```

---

simulate_design *Simulate a selected design*

Description

Estimates operating characteristics by Monte Carlo simulation for one or more true response rates.

Usage

```
simulate_design(design, true_rates, n_sim = 10000L, seed = NULL)
```

Arguments

design	A pob2ef_design object returned by select_design().
true_rates	True response rates to simulate.
n_sim	Number of simulated trials per true response rate.
seed	Optional integer random-number seed.

Value

An object of class `pob2ef_simulation` containing scenario-level operating-characteristic estimates and Monte Carlo standard errors.

Examples

```
design <- select_design(
  n_max = 20, looks = c(10, 20), p0 = 0.20, p1 = 0.40,
  alpha = 0.10, beta = 0.20,
  lambda1_grid = 0.50, lambda2_grid = 0.20, gamma_grid = 0, eta_grid = 0
)
simulation <- simulate_design(
  design, true_rates = c(0.20, 0.30, 0.40), n_sim = 100, seed = 2026
)
simulation$results
```

Index

decision_table, 2

plot_decision_boundaries, 2
plot_operating_characteristics, 3

select_design, 4
select_design(), *2*, *3*, *5*
simulate_design, 5
simulate_design(), *3*