Package 'AdaptiveTrialsR'

July 30, 2026

Type Package

Title Bayesian and Adaptive Clinical Trial Simulation and Monitoring

Version 0.1.2

Description Provides tools for simulating, monitoring, and evaluating Bayesian and adaptive clinical trial designs with binary outcomes. The package includes methods for trial simulation, posterior and predictive probability monitoring, adaptive randomization, Bayesian treatment-effect estimation, dose-finding designs, two-stage phase II trial designs, group sequential monitoring, operating-characteristic evaluation, interim analyses, and decision-support procedures. The methodological foundation includes the two-stage phase II design described by Simon (1989) <doi:10.1016/0197-2456(89)90015-9>.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

Imports stats, utils, dplyr, ggplot2, rlang

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

URL https://github.com/zerish12/AdaptiveTrialsR

BugReports https://github.com/zerish12/AdaptiveTrialsR/issues

NeedsCompilation no

Author Muhammad Zahir Khan [aut, cre]

Maintainer Muhammad Zahir Khan <zahirstat007@gmail.com>

Repository CRAN

Date/Publication 2026-07-30 17:30:21 UTC

Contents

`adaptive_randomisation` *Adaptive Randomisation Probabilities*

Description

Calculates response-adaptive randomisation probabilities from posterior mean response rates. Better-performing arms receive higher allocation probabilities while preserving minimum and maximum allocation limits.

Usage

```
adaptive_randomisation(
  posterior_means,
  min_prob = 0.1,
  max_prob = 0.8,
  power = 1
)
```

Arguments

posterior_means	Numeric vector of posterior mean response rates. Names may be supplied for treatment arms.
min_prob	Minimum allocation probability for each arm. Default is 0.10.
max_prob	Maximum allocation probability for any arm. Default is 0.80.
power	Tuning parameter controlling how strongly allocation favours better-performing arms. Default is 1.

Value

A data frame with treatment arms, posterior means, and allocation probabilities.

Examples

```
adaptive_randomisation(
  posterior_means = c(Control = 0.30, Treatment = 0.50)
)

adaptive_randomisation(
  posterior_means = c(Control = 0.25, TreatmentA = 0.45, TreatmentB = 0.60),
  min_prob = 0.10,
  max_prob = 0.80
)
```

---

analyse_trial_data *Analyse Real Trial Data Using Bayesian Monitoring*

Description

Applies Bayesian posterior monitoring to real patient-level binary-outcome clinical trial data. The function compares a treatment arm with a control arm using a beta-binomial model.

Usage

```
analyse_trial_data(
  data,
  arm_col = "arm",
  outcome_col = "outcome",
  control_arm,
  treatment_arm,
  margin = 0,
  efficacy_threshold = 0.95,
  futility_threshold = 0.1,
  prior_alpha = 1,
  prior_beta = 1,
  nsim = 10000,
  seed = NULL
)
```

Arguments

`data`	A data frame containing patient-level trial data.
`arm_col`	Name of the treatment-arm column.
`outcome_col`	Name of the binary outcome column.
`control_arm`	Name of the control arm.
`treatment_arm`	Name of the treatment arm.
`margin`	Minimum clinically important treatment-control difference.
`efficacy_threshold`	Posterior probability threshold for declaring efficacy.
`futility_threshold`	Posterior probability threshold for futility.
`prior_alpha`	Beta prior alpha parameter.
`prior_beta`	Beta prior beta parameter.
`nsim`	Number of posterior simulations.
`seed`	Optional random seed.

Value

A data frame containing arm-level summaries, posterior probability, and Bayesian decision.

Examples

```
trial_data <- data.frame(
  arm = c("Control", "Control", "Control", "Treatment", "Treatment", "Treatment"),
  outcome = c(0, 1, 0, 1, 1, 0)
)

analyse_trial_data(
  data = trial_data,
```

```
  control_arm = "Control",
  treatment_arm = "Treatment",
  seed = 123
)
```

`compare_trial_scenarios` *Compare Operating Characteristics Across Trial Scenarios*

Description

Runs repeated Bayesian adaptive trial simulations across multiple true-response scenarios and summarises the resulting operating characteristics.

Usage

```
compare_trial_scenarios(
  scenarios,
  n_sim = 100,
  n_total = 100,
  interim_every = 20,
  margin = 0,
  efficacy_threshold = 0.95,
  futility_threshold = 0.1,
  prior_alpha = 1,
  prior_beta = 1,
  nsim = 10000,
  seed = NULL
)
```

Arguments

`scenarios`	A data frame with columns control_response and treatment_response.
`n_sim`	Number of simulated trials per scenario.
`n_total`	Maximum total sample size per trial.
`interim_every`	Frequency of interim analyses.
`margin`	Minimum clinically important treatment-control difference.
`efficacy_threshold`	Posterior probability threshold for stopping early for efficacy.
`futility_threshold`	Posterior probability threshold for stopping early for futility.
`prior_alpha`	Beta prior alpha parameter.
`prior_beta`	Beta prior beta parameter.
`nsim`	Number of posterior simulations used inside each trial.
`seed`	Optional random seed.

Value

A data frame of operating characteristics by scenario.

Examples

```
scenarios <- data.frame(
  control_response = c(0.25, 0.25, 0.25),
  treatment_response = c(0.25, 0.35, 0.45)
)

compare_trial_scenarios(
  scenarios = scenarios,
  n_sim = 10,
  n_total = 100,
  interim_every = 20,
  nsim = 1000,
  seed = 123
)
```

---

create_trial_template *Create a Trial Data Template*

Description

Creates a simple patient-level clinical trial data template for use with AdaptiveTrialsR. The template includes patient ID, treatment arm, and binary outcome columns.

Usage

```
create_trial_template(n = 20, arms = c("Control", "Treatment"), path = NULL)
```

Arguments

n	A single positive integer giving the number of rows to create. The default is 20.
arms	A character vector containing at least two non-empty treatment arm names. The default is c("Control", "Treatment").
path	Either NULL or a single non-empty character string specifying the complete CSV output file path. When NULL, no file is written.

Details

No file is written unless path is explicitly supplied. Package examples and tests that write files should use tempfile() or tempdir().

Value

A data frame containing the trial data template.

Examples

```
template <- create_trial_template(n = 10)
head(template)

temp_file <- tempfile(fileext = ".csv")

template <- create_trial_template(
  n = 10,
  path = temp_file
)

file.exists(temp_file)
unlink(temp_file)
```

`crm_dose_finding` *Simple Bayesian CRM Dose-Finding Summary*

Description

Provides a simplified Bayesian continual reassessment method-style dose-finding summary for Phase I trials. The function estimates posterior toxicity probabilities for each dose and recommends the dose closest to the target toxicity rate.

Usage

```
crm_dose_finding(
  dose_data,
  dose_col = "dose",
  n_col = "n",
  tox_col = "tox",
  target_toxicity = 0.25,
  prior_alpha = 1,
  prior_beta = 1
)
```

Arguments

`dose_data`	A data frame containing dose-level data.
`dose_col`	Name of the dose column.
`n_col`	Name of the column containing number treated.
`tox_col`	Name of the column containing number with toxicity.
`target_toxicity`	Target toxicity probability.
`prior_alpha`	Beta prior alpha parameter.
`prior_beta`	Beta prior beta parameter.

Value

A data frame with posterior toxicity summaries and recommended dose.

Examples

```
dose_data <- data.frame(
  dose = c(1, 2, 3, 4),
  n = c(3, 3, 3, 3),
  tox = c(0, 0, 1, 2)
)

crm_dose_finding(dose_data)
```

`decision_summary` *Summarise Final Decisions from Adaptive Trial Simulations*

Description

Creates a clean summary table of final trial decisions from repeated Bayesian adaptive clinical trial simulations.

Usage

```
decision_summary(simulations)
```

Arguments

`simulations`	An object returned by `simulate_trials()`.

Value

A data frame containing decision counts and percentages.

Examples

```
sims <- simulate_trials(
  n_sim = 20,
  n_total = 100,
  true_response = c(0.25, 0.45),
  interim_every = 20,
  nsim = 1000,
  seed = 123
)

decision_summary(sims)
```

`estimate_treatment_effect` *Estimate Bayesian Treatment Effect for Binary Trial Outcomes*

Description

Estimates the posterior distribution of the treatment-control difference in response probabilities using a beta-binomial model.

Usage

```
estimate_treatment_effect(
  data,
  arm_col = "arm",
  outcome_col = "outcome",
  control_arm,
  treatment_arm,
  margin = 0,
  efficacy_threshold = 0.95,
  futility_threshold = 0.1,
  prior_alpha = 1,
  prior_beta = 1,
  nsim = 10000,
  cred_level = 0.95,
  seed = NULL
)
```

Arguments

`data`	A data frame containing patient-level trial data.
`arm_col`	Name of the treatment-arm column.
`outcome_col`	Name of the binary outcome column.
`control_arm`	Name of the control arm.
`treatment_arm`	Name of the treatment arm.
`margin`	Minimum clinically important treatment-control difference.
`efficacy_threshold`	Posterior probability threshold for declaring efficacy.
`futility_threshold`	Posterior probability threshold for futility.
`prior_alpha`	Beta prior alpha parameter.
`prior_beta`	Beta prior beta parameter.
`nsim`	Number of posterior simulations.
`cred_level`	Credible interval level.
`seed`	Optional random seed.

Value

A data frame containing posterior treatment-effect summaries.

Examples

```
trial_data <- data.frame(
  arm = c("Control", "Control", "Treatment", "Treatment"),
  outcome = c(0, 1, 1, 1)
)

estimate_treatment_effect(
  data = trial_data,
  control_arm = "Control",
  treatment_arm = "Treatment",
  seed = 123
)
```

---

export_figures_to_desktop

Export a Figures Directory

Description

Copies a directory containing figures to a destination explicitly selected by the user.

Usage

```
export_figures_to_desktop(
  figures_dir,
  destination_dir,
  folder_name = NULL,
  overwrite = FALSE,
  verbose = FALSE
)
```

Arguments

`figures_dir`	A single non-empty character string specifying the existing source directory containing the figures to copy.
`destination_dir`	A single non-empty character string specifying the existing parent directory into which the figures directory will be copied.
`folder_name`	A single non-empty character string specifying the name of the copied directory. If `NULL`, the base name of `figures_dir` is used.
`overwrite`	Logical. If `TRUE`, an existing destination directory is removed before the files are copied. The default is `FALSE`.
`verbose`	Logical. If `TRUE`, the final destination path is displayed. The default is `FALSE`.

Details

The function does not select the Desktop, home directory, package directory, or current working directory automatically. Both the source directory and destination directory must be supplied explicitly.

This function writes files only after both `figures_dir` and `destination_dir` have been explicitly supplied.

Package examples and tests should use `tempfile()` or `tempdir()` for all file-writing operations.

Value

Invisibly returns the normalized path of the copied directory.

Examples

```
source_dir <- tempfile("AdaptiveTrialsR-source-")
destination_dir <- tempfile("AdaptiveTrialsR-destination-")

dir.create(source_dir)
dir.create(destination_dir)

example_file <- file.path(source_dir, "example-figure.txt")
writeLines("Example figure placeholder", con = example_file)

copied_path <- export_figures_to_desktop(
  figures_dir = source_dir,
  destination_dir = destination_dir,
  folder_name = "copied-figures"
)

dir.exists(copied_path)

unlink(source_dir, recursive = TRUE, force = TRUE)
unlink(destination_dir, recursive = TRUE, force = TRUE)
```

`export_trial_results` *Export Adaptive Trial Simulation Results*

Description

Exports simulation-level results, operating characteristics, and decision summaries from repeated Bayesian adaptive trial simulations to CSV files.

Usage

```
export_trial_results(simulations, path, prefix = "adaptive_trial")
```

Arguments

simulations	An object returned by simulate_trials().
path	A single non-empty character string specifying the existing directory where the CSV files should be saved. This argument must be supplied explicitly.
prefix	A single non-empty character string used as the prefix for exported file names. The default is "adaptive_trial".

Details

The function writes files only after path has been explicitly supplied. It does not write by default to the current working directory, package directory, home directory, or Desktop.

In package examples and tests, temporary directories should be used.

Value

Invisibly returns a named character vector containing the normalized paths of the three written CSV files.

Examples

```
sims <- simulate_trials(
  n_sim = 10,
  n_total = 100,
  true_response = c(0.25, 0.45),
  interim_every = 20,
  nsim = 1000,
  seed = 123
)

output_dir <- tempfile("AdaptiveTrialsR-results-")
dir.create(output_dir)

written_files <- export_trial_results(
  simulations = sims,
  path = output_dir
)

file.exists(written_files)

unlink(output_dir, recursive = TRUE, force = TRUE)
```

---

group_sequential_boundary

Group Sequential Monitoring Boundaries

Description

Creates simple group sequential monitoring boundaries for interim analyses in a clinical trial. The function provides approximate efficacy and futility boundaries across planned information fractions.

Usage

```
group_sequential_boundary(
  looks = 5,
  alpha = 0.05,
  beta = 0.2,
  boundary = "obrien_fleming"
)
```

Arguments

looks	Number of interim looks including the final analysis.
alpha	Overall type I error rate. Default is 0.05.
beta	Type II error rate. Default is 0.20.
boundary	Type of efficacy boundary. Options are "obrien_fleming" and "pocock".

Value

A data frame containing information fractions and monitoring boundaries.

Examples

```
group_sequential_boundary(
  looks = 5,
  alpha = 0.05,
  beta = 0.20,
  boundary = "obrien_fleming"
)
```

---

interim_monitoring *Bayesian Interim Monitoring for Adaptive Clinical Trials*

Description

Performs Bayesian interim monitoring using a beta-binomial model and calculates the posterior probability that the treatment response rate exceeds the control response rate by a specified margin.

Usage

```
interim_monitoring(
  y_treatment,
  n_treatment,
  y_control,
  n_control,
  margin = 0,
  efficacy_threshold = 0.95,
  futility_threshold = 0.1,
  prior_alpha = 1,
```

```
  prior_beta = 1,
  nsim = 10000,
  seed = NULL
)
```

Arguments

y_treatment	Number of responders in the treatment arm.
n_treatment	Number of patients in the treatment arm.
y_control	Number of responders in the control arm.
n_control	Number of patients in the control arm.
margin	Minimum clinically important difference. Default is 0.
efficacy_threshold	Posterior probability threshold for stopping early for efficacy. Default is 0.95.
futility_threshold	Posterior probability threshold for stopping early for futility. Default is 0.10.
prior_alpha	Beta prior alpha parameter. Default is 1.
prior_beta	Beta prior beta parameter. Default is 1.
nsim	Number of posterior simulations. Default is 10000.
seed	Optional random seed.

Value

A data frame containing interim monitoring results.

Examples

```
interim_monitoring(
  y_treatment = 18,
  n_treatment = 40,
  y_control = 10,
  n_control = 40
)
```

---

plot_dose_finding *Plot Bayesian Dose-Finding Results*

Description

Creates a publication-quality plot of posterior toxicity probabilities across dose levels from a Bayesian CRM-style dose-finding summary.

Usage

```
plot_dose_finding(
  dose_result,
  title = "Bayesian Phase I Dose-Finding Summary",
  subtitle = "Posterior toxicity estimates by dose level"
)
```

Arguments

dose_result	A data frame returned by crm_dose_finding().
title	Plot title.
subtitle	Plot subtitle.

Value

A ggplot object.

Examples

```
dose_data <- data.frame(
  dose = c(1, 2, 3, 4),
  n = c(3, 3, 3, 3),
  tox = c(0, 0, 1, 2)
)

dose_result <- crm_dose_finding(dose_data)
plot_dose_finding(dose_result)
```

---

plot_group_sequential_boundary

Plot Group Sequential Monitoring Boundaries

Description

Creates a publication-quality plot of efficacy and futility monitoring boundaries across planned interim analyses.

Usage

```
plot_group_sequential_boundary(
  boundary_data,
  title = "Group Sequential Monitoring Boundaries",
  subtitle = "Approximate efficacy and futility boundaries across interim looks"
)
```

Arguments

boundary_data	A data frame returned by group_sequential_boundary().
title	Plot title.
subtitle	Plot subtitle.

Value

A ggplot object.

Examples

```
boundaries <- group_sequential_boundary(
  looks = 5,
  alpha = 0.05,
  beta = 0.20
)

plot_group_sequential_boundary(boundaries)
```

---

plot_operating_characteristics

Plot Operating Characteristics from Simulated Adaptive Trials

---

Description

Creates a publication-quality bar plot of key operating characteristics from repeated Bayesian adaptive clinical trial simulations.

Usage

```
plot_operating_characteristics(
  simulations,
  title = "Operating Characteristics of Bayesian Adaptive Trial Design",
  subtitle = "Estimated from repeated trial simulations"
)
```

Arguments

simulations	An object returned by simulate_trials().
title	Plot title.
subtitle	Plot subtitle.

Value

A ggplot object showing key operating characteristics.

Examples

```
sims <- simulate_trials(
  n_sim = 10,
  n_total = 100,
  true_response = c(0.25, 0.45),
  interim_every = 20,
  nsim = 1000,
  seed = 123
)

plot_operating_characteristics(sims)
```

`plot_posterior_density` *Plot Posterior Density for Trial Arms*

Description

Creates a publication-quality density plot of posterior response probabilities for each trial arm using a beta-binomial model.

Usage

```
plot_posterior_density(
  data,
  arm_col = "arm",
  outcome_col = "outcome",
  prior_alpha = 1,
  prior_beta = 1,
  nsim = 10000,
  seed = NULL,
  title = "Posterior Response Distributions by Trial Arm",
 subtitle = "Beta-binomial posterior densities for binary clinical trial outcomes"
)
```

Arguments

`data`	A data frame containing patient-level trial data.
`arm_col`	Name of the treatment-arm column.
`outcome_col`	Name of the binary outcome column.
`prior_alpha`	Beta prior alpha parameter.
`prior_beta`	Beta prior beta parameter.
`nsim`	Number of posterior simulations.
`seed`	Optional random seed.
`title`	Plot title.
`subtitle`	Plot subtitle.

Value

A ggplot object.

Examples

```
trial_data <- data.frame(
  arm = c("Control", "Control", "Treatment", "Treatment"),
  outcome = c(0, 1, 1, 1)
)

plot_posterior_density(trial_data, seed = 123)
```

---

plot_posterior_intervals

Plot Posterior Credible Intervals for Trial Arms

Description

Creates a publication-quality forest-style plot of posterior mean response rates and credible intervals for each trial arm.

Usage

```
plot_posterior_intervals(
  posterior_data,
  title = "Posterior Response Estimates by Trial Arm",
  subtitle = "Posterior means with 95% credible intervals"
)
```

Arguments

`posterior_data`	A data frame returned by `posterior_summary()`.
`title`	Plot title.
`subtitle`	Plot subtitle.

Value

A ggplot object.

Examples

```
trial_data <- data.frame(
  arm = c("Control", "Control", "Treatment", "Treatment"),
  outcome = c(0, 1, 1, 1)
)

post <- posterior_summary(trial_data)
plot_posterior_intervals(post)
```

plot_real_trial_results *Plot Real Trial Analysis Results*

Description

Creates a publication-quality bar plot comparing observed response rates between control and treatment arms from real patient-level trial data.

Usage

```
plot_real_trial_results(
  analysis_result,
  title = "Bayesian Analysis of Real Trial Data",
  subtitle = "Observed response rates by treatment arm"
)
```

Arguments

analysis_result	A data frame returned by analyse_trial_data().
title	Plot title.
subtitle	Plot subtitle.

Value

A ggplot object.

Examples

```
trial_data <- data.frame(
  arm = c("Control", "Control", "Control", "Treatment", "Treatment", "Treatment"),
  outcome = c(0, 1, 0, 1, 1, 0)
)

result <- analyse_trial_data(
  data = trial_data,
  control_arm = "Control",
  treatment_arm = "Treatment",
  seed = 123
)

plot_real_trial_results(result)
```

`plot_sample_size_distribution` *Plot Sample Size Distribution from Adaptive Trial Simulations*

Description

Creates a publication-quality histogram showing the distribution of enrolled sample sizes across repeated Bayesian adaptive clinical trial simulations.

Usage

```
plot_sample_size_distribution(
  simulations,
  title = "Sample Size Distribution Across Adaptive Trial Simulations",
  subtitle =
   "Distribution of enrolled patients before stopping or reaching maximum sample size"
)
```

Arguments

`simulations`	An object returned by `simulate_trials()`.
`title`	Plot title.
`subtitle`	Plot subtitle.

Value

A ggplot object.

Examples

```
sims <- simulate_trials(
  n_sim = 20,
  n_total = 100,
  true_response = c(0.25, 0.45),
  interim_every = 20,
  nsim = 1000,
  seed = 123
)

plot_sample_size_distribution(sims)
```

plot_scenario_comparison *Plot Scenario Comparison for Adaptive Trial Simulations*

Description

Creates a publication-quality plot comparing operating characteristics across multiple true treatment-effect scenarios.

Usage

```
plot_scenario_comparison(
  scenario_results,
  metric = "probability_stop_efficacy",
  title = "Operating Characteristics Across Trial Scenarios",
 subtitle = "Bayesian adaptive trial performance under different treatment effects"
)
```

Arguments

scenario_results	A data frame returned by compare_trial_scenarios().
metric	Operating characteristic to plot. Default is "probability_stop_efficacy".
title	Plot title.
subtitle	Plot subtitle.

Value

A ggplot object.

Examples

```
scenarios <- data.frame(
  control_response = c(0.25, 0.25, 0.25),
  treatment_response = c(0.25, 0.35, 0.45)
)

scenario_results <- compare_trial_scenarios(
  scenarios = scenarios,
  n_sim = 10,
  n_total = 100,
  interim_every = 20,
  nsim = 1000,
  seed = 123
)

plot_scenario_comparison(scenario_results)
```

`plot_treatment_effect` *Plot Bayesian Treatment Effect*

Description

Creates a publication-quality plot of the posterior treatment-control difference with a Bayesian credible interval.

Usage

```
plot_treatment_effect(
  effect_result,
  title = "Bayesian Treatment Effect Estimate",
  subtitle = "Posterior treatment-control difference with credible interval"
)
```

Arguments

`effect_result`	A data frame returned by `estimate_treatment_effect()`.
`title`	Plot title.
`subtitle`	Plot subtitle.

Value

A ggplot object.

Examples

```
trial_data <- data.frame(
  arm = c("Control", "Control", "Treatment", "Treatment"),
  outcome = c(0, 1, 1, 1)
)

effect <- estimate_treatment_effect(
  data = trial_data,
  control_arm = "Control",
  treatment_arm = "Treatment",
  seed = 123
)

plot_treatment_effect(effect)
```

plot_trial_path *Plot Interim Monitoring Path for an Adaptive Trial*

Description

Creates a publication-quality plot of posterior probability across interim analyses for a simulated Bayesian adaptive clinical trial.

Usage

```
plot_trial_path(
  trial,
  title = "Bayesian Interim Monitoring Path",
  subtitle = "Posterior probability of treatment benefit across interim analyses"
)
```

Arguments

trial	An object returned by simulate_trial().
title	Plot title.
subtitle	Plot subtitle.

Value

A ggplot object showing the Bayesian interim monitoring path.

Examples

```
trial <- simulate_trial(
  n_total = 100,
  true_response = c(0.25, 0.45),
  interim_every = 20,
  seed = 123
)

plot_trial_path(trial)
```

posterior_summary *Posterior Summary for Binary Trial Outcomes*

Description

Computes Bayesian posterior summaries for binary response rates in each treatment arm using a beta-binomial model.

Usage

```
posterior_summary(
  data,
  arm_col = "arm",
  outcome_col = "outcome",
  prior_alpha = 1,
  prior_beta = 1,
  cred_level = 0.95
)
```

Arguments

data	A data frame containing patient-level trial data.
arm_col	Name of the treatment-arm column.
outcome_col	Name of the binary outcome column.
prior_alpha	Beta prior alpha parameter.
prior_beta	Beta prior beta parameter.
cred_level	Credible interval level. Default is 0.95.

Value

A data frame containing posterior summaries by treatment arm.

Examples

```
trial_data <- data.frame(
  arm = c("Control", "Control", "Treatment", "Treatment"),
  outcome = c(0, 1, 1, 1)
)

posterior_summary(trial_data)
```

---

predictive_probability

Predictive Probability of Trial Success

Description

Estimates the predictive probability that a trial will meet its final success criterion if recruitment continues to the planned maximum sample size. This is useful for Bayesian futility monitoring in adaptive trials.

Usage

```
predictive_probability(
  y_treatment,
  n_treatment,
  y_control,
  n_control,
  n_total,
  margin = 0,
  success_threshold = 0.95,
  prior_alpha = 1,
  prior_beta = 1,
  nsim = 10000,
  seed = NULL
)
```

Arguments

y_treatment	Current number of responders in the treatment arm.
n_treatment	Current number of patients in the treatment arm.
y_control	Current number of responders in the control arm.
n_control	Current number of patients in the control arm.
n_total	Planned total sample size across both arms.
margin	Minimum clinically important treatment-control difference.
success_threshold	Posterior probability threshold for final success.
prior_alpha	Beta prior alpha parameter.
prior_beta	Beta prior beta parameter.
nsim	Number of simulation draws.
seed	Optional random seed.

Value

A data frame containing predictive probability and decision.

Examples

```
predictive_probability(
  y_treatment = 12,
  n_treatment = 30,
  y_control = 8,
  n_control = 30,
  n_total = 100,
  seed = 123
)
```

`predict_success` *Predict Bayesian Probability of Trial Success*

Description

Calculates the posterior probability that the treatment response rate exceeds the control response rate by at least a specified margin using a beta-binomial model.

Usage

```
predict_success(
  y_treatment,
  n_treatment,
  y_control,
  n_control,
  margin = 0,
  prior_alpha = 1,
  prior_beta = 1,
  nsim = 10000,
  seed = NULL
)
```

Arguments

`y_treatment`	Number of responders in the treatment arm.
`n_treatment`	Number of patients in the treatment arm.
`y_control`	Number of responders in the control arm.
`n_control`	Number of patients in the control arm.
`margin`	Minimum clinically important difference. Default is 0.
`prior_alpha`	Beta prior alpha parameter. Default is 1.
`prior_beta`	Beta prior beta parameter. Default is 1.
`nsim`	Number of posterior simulations. Default is 10000.
`seed`	Optional random seed.

Value

A data frame containing posterior probability and decision.

Examples

```
predict_success(
  y_treatment = 18,
  n_treatment = 40,
  y_control = 10,
  n_control = 40
)
```

`sample_size_binary` *Sample Size for Two-Arm Binary Endpoint Trial*

Description

Calculates an approximate sample size for a two-arm clinical trial with binary outcomes using a normal approximation for comparing two proportions.

Usage

```
sample_size_binary(
  p_control,
  p_treatment,
  alpha = 0.05,
  power = 0.8,
  allocation_ratio = 1
)
```

Arguments

`p_control`	Expected response probability in the control arm.
`p_treatment`	Expected response probability in the treatment arm.
`alpha`	Type I error rate. Default is 0.05.
`power`	Desired statistical power. Default is 0.80.
`allocation_ratio`	Allocation ratio treatment/control. Default is 1.

Value

A data frame containing sample size calculations.

Examples

```
sample_size_binary(
  p_control = 0.25,
  p_treatment = 0.45
)
```

simon_two_stage *Simon Two-Stage Phase II Trial Design*

Description

Searches for a Simon two-stage design for a single-arm phase II trial with a binary response outcome. The design compares an unacceptable response probability under the null hypothesis with a desirable response probability under the alternative hypothesis while controlling the type I error rate and achieving the requested statistical power.

Usage

```
simon_two_stage(p0, p1, alpha = 0.05, power = 0.8, n_max = 100)
```

Arguments

p0	Numeric value giving the unacceptable response probability under the null hypothesis. Must lie strictly between 0 and 1.
p1	Numeric value giving the desirable response probability under the alternative hypothesis. Must be greater than p0 and strictly less than 1.
alpha	Numeric value giving the maximum allowable type I error probability. Default is 0.05.
power	Numeric value giving the minimum required statistical power. Default is 0.80.
n_max	Positive integer giving the maximum total sample size included in the search. Default is 100.

Details

The trial stops after stage 1 for lack of activity when the observed number of responses is less than or equal to r_stage1. Otherwise, the second stage is conducted. Candidate designs satisfying the requested type I error and power constraints are ranked first by expected sample size under the null hypothesis and then by maximum total sample size.

Value

A one-row data frame describing the selected two-stage design, containing:

- n_stage1: first-stage sample size;
- n_stage2: second-stage sample size;
- n_total: maximum total sample size;
- r_stage1: early-termination boundary after stage 1;
- r_total: minimum total response boundary used by the design;
- p0 and p1: response probabilities used in the search;
- type1_error: achieved type I error probability;

- `power`: achieved statistical power;
- `probability_early_termination_p0`: probability of early termination under the null hypothesis;
- `expected_sample_size_p0`: expected sample size under the null;
- `design_type`: criterion used to select the reported design.

References

Simon, R. (1989). Optimal two-stage designs for phase II clinical trials. *Controlled Clinical Trials*, 10(1), 1–10. doi:10.1016/01972456(89)900159

Examples

```
design <- simon_two_stage(
  p0 = 0.20,
  p1 = 0.40,
  alpha = 0.20,
  power = 0.60,
  n_max = 15
)

design
```

---

simulate_trial *Simulate a Bayesian Adaptive Clinical Trial*

Description

Simulates a two-arm Bayesian adaptive clinical trial with binary outcomes. The trial can include interim monitoring for early stopping based on posterior probability thresholds.

Usage

```
simulate_trial(
  n_total = 100,
  true_response = c(0.25, 0.4),
  arm_names = c("Control", "Treatment"),
  interim_every = 20,
  margin = 0,
  efficacy_threshold = 0.95,
  futility_threshold = 0.1,
  prior_alpha = 1,
  prior_beta = 1,
  nsim = 10000,
  seed = NULL
)
```

Arguments

n_total	Maximum total sample size.
true_response	Numeric vector of true response probabilities for control and treatment arms.
arm_names	Character vector of treatment arm names.
interim_every	Frequency of interim analyses.
margin	Minimum clinically important treatment-control difference.
efficacy_threshold	Posterior probability threshold for stopping early for efficacy.
futility_threshold	Posterior probability threshold for stopping early for futility.
prior_alpha	Beta prior alpha parameter.
prior_beta	Beta prior beta parameter.
nsim	Number of posterior simulations.
seed	Optional random seed.

Value

A list containing patient-level data, interim results, and final trial decision.

Examples

```
simulate_trial(
  n_total = 100,
  true_response = c(0.25, 0.40),
  interim_every = 20,
  seed = 123
)
```

---

simulate_trials *Simulate Multiple Bayesian Adaptive Clinical Trials*

Description

Runs repeated simulations of a Bayesian adaptive clinical trial and estimates key operating characteristics, including stopping probability, efficacy stopping rate, futility stopping rate, mean sample size, and probability of trial success.

Usage

```
simulate_trials(
  n_sim = 100,
  n_total = 100,
  true_response = c(0.25, 0.4),
  arm_names = c("Control", "Treatment"),
```

```
  interim_every = 20,
  margin = 0,
  efficacy_threshold = 0.95,
  futility_threshold = 0.1,
  prior_alpha = 1,
  prior_beta = 1,
  nsim = 10000,
  seed = NULL
)
```

Arguments

n_sim	Number of simulated trials.
n_total	Maximum total sample size per trial.
true_response	Numeric vector of true response probabilities for control and treatment arms.
arm_names	Character vector of treatment arm names.
interim_every	Frequency of interim analyses.
margin	Minimum clinically important treatment-control difference.
efficacy_threshold	Posterior probability threshold for stopping early for efficacy.
futility_threshold	Posterior probability threshold for stopping early for futility.
prior_alpha	Beta prior alpha parameter.
prior_beta	Beta prior beta parameter.
nsim	Number of posterior simulations used inside each trial.
seed	Optional random seed.

Value

A list containing simulation-level results and operating characteristics.

Examples

```
sims <- simulate_trials(
  n_sim = 10,
  n_total = 100,
  true_response = c(0.25, 0.45),
  interim_every = 20,
  nsim = 1000,
  seed = 123
)

sims$operating_characteristics
```

summary_trial *Summarise a Simulated Adaptive Trial*

Description

Provides a concise summary of a simulated adaptive clinical trial object produced by `simulate_trial()`.

Usage

```
summary_trial(trial)
```

Arguments

`trial`	An object returned by `simulate_trial()`.

Value

A data frame summarising trial sample size, responses, stopping status, and final decision.

Examples

```
trial <- simulate_trial(
  n_total = 100,
  true_response = c(0.25, 0.40),
  interim_every = 20,
  seed = 123
)

summary_trial(trial)
```

trial_report *Create a Trial Report*

Description

Creates a trial report object and optionally writes a text representation of the report to a file explicitly selected by the user.

Usage

```
trial_report(..., file = NULL, verbose = FALSE)
```

Arguments

...	Additional arguments used to calculate the trial report.
file	Either NULL or a single non-empty character string specifying the complete output file path. When NULL, no file is written.
verbose	Logical. If TRUE, a message showing the written file path is displayed. The default is FALSE.

Details

No file is written unless file is explicitly supplied by the user. Package examples and tests that write files should use tempfile() or tempdir().

Value

A list containing the calculated trial report.

Examples

```
report <- trial_report()

output_file <- tempfile(fileext = ".txt")

report <- trial_report(
  file = output_file,
  verbose = FALSE
)

file.exists(output_file)
unlink(output_file)
```

---

validate_trial_data *Validate Real Trial Data for Adaptive Trial Analysis*

Description

Checks whether a real clinical trial dataset has the minimum required structure for use with AdaptiveTrialsR functions. The dataset must contain one row per patient, a treatment-arm variable, and a binary outcome variable.

Usage

```
validate_trial_data(
  data,
  arm_col = "arm",
  outcome_col = "outcome",
  allowed_outcomes = c(0, 1)
)
```

Arguments

`data`	A data frame containing patient-level trial data.
`arm_col`	Name of the treatment-arm column.
`outcome_col`	Name of the binary outcome column.
`allowed_outcomes`	Allowed values for the binary outcome. Default is c(0, 1).

Value

A cleaned data frame with standardised columns `arm` and `outcome`.

Examples

```
trial_data <- data.frame(
  patient_id = 1:6,
  arm = c("Control", "Control", "Treatment", "Treatment", "Control", "Treatment"),
  outcome = c(0, 1, 1, 1, 0, 1)
)

validate_trial_data(trial_data)
```

Index