Package 'rsmart'

September 24, 2026

Title Sequential Multiple Assignment Randomized Trials Design and Analyses

Version 0.1.0

Description Implements the interim augmented inverse probability weighted estimator (IAIPWE) for estimating the value of treatment regimes embedded in sequential multiple assignment randomized trials (SMARTs). The IAIPWE subsumes both the inverse probability weighted estimator (IPWE) and the augmented inverse probability weighted estimator (AIPWE), enabling inference at interim and final analyses. The package provides functions for value estimation, sandwich variance computation, group sequential stopping boundaries, and sample size determination for multi-stage SMARTs with up to two treatment options at each stage. See Manschot, Laber, and Davidian (2023) <doi:10.1111/biom.13854> for additional details.

License GPL (>= 3)

URL https://MSDLLCpapers.github.io/rsmart/, https://github.com/MSDLLCpapers/rsmart, https://msdllcpapers.github.io/rsmart/

BugReports https://github.com/MSDLLCpapers/rsmart/issues

Encoding UTF-8

RoxygenNote 7.3.3

Suggests dplyr, knitr, pkgdown, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

Depends R (>= 3.5)

Imports data.table, doFuture, foreach, Matrix, MASS, modelObj, mvtnorm, stats

LazyData true

VignetteBuilder knitr

NeedsCompilation no

Author Cole Manschot [aut, cre],
Tabitha Peter [ctb],
Gabriela Piasecki [ctb],
Laura Frederick [ctb]

Maintainer Cole Manschot <cole.manschot@msd.com>

Repository CRAN

Date/Publication 2026-09-24 13:40:02 UTC

Contents

`block_rand` *Create a blocked randomization function*

Description

Returns a randomization function compatible with `sim_treatment` that implements permuted block randomization. Within each block, every treatment appears exactly `block_rep` times, and the order within blocks is randomly permuted. This ensures approximate balance across treatments at any point during enrollment.

Usage

```
block_rand(block_rep = 2)
```

Arguments

block_rep	A positive integer. Number of times each treatment appears in every block. The block size is block_rep * n_treatments. Default is 2.

Value

A function with signature function(n, n_treatments, prob) suitable for use as rand_prob_fn in sim_treatment. The prob argument is accepted but ignored, since blocked randomization enforces equal allocation within each block.

Examples

```
# Use blocked randomization with sim_treatment
set.seed(1)
a <- sim_treatment(n = 100, n_treatments = 2,
                   rand_prob_fn = block_rand(block_rep = 2))
table(a)

# Three treatments with block size 6 (block_rep = 2)
set.seed(1)
a <- sim_treatment(n = 300, n_treatments = 3,
                   rand_prob_fn = block_rand(block_rep = 2))
table(a)

# Stage 2 blocked randomization within prior treatment groups
set.seed(1)
df <- data.frame(a1 = c(rep(0, 50), rep(1, 50)))
df <- sim_treatment(n_treatments = 2, dat = df, stage = 2,
                    rand_prob_fn = block_rand(block_rep = 3))
table(df$a1, df$a2)
```

---

estimate_values *Estimate values for all treatment regimes*

Description

Wraps the outcome regression and value term functions to estimate the value of all regimes in the provided list. For each regime, Q-functions are fitted (if specified), augmentation and IPW terms are computed, and the regime value is estimated.

Usage

```
estimate_values(df, q_list, regime_all, feasible_sets_indicator, pis, nus)
```

Arguments

df	A data frame containing the trial data, including treatment assignments, covariates, outcomes, and a kappa column.
q_list	A list of outcome regression model specifications (one per stage), or NULL for IPW estimation.
regime_all	A list of regime objects, each containing a regime matrix and a regime_ind indicator matrix.
feasible_sets_indicator	A logical value indicating whether feasible sets are present in the trial design.
pis	A data frame of estimated propensity scores with columns pi1, ..., piK.
nus	A list as returned by get_nu, containing the estimated stage probabilities.

Value

A list with the following components:

value A numeric vector of estimated regime values.

df A list of matrices of value term components, augmentation terms 1 through 2K then the IPW term for each regime.

q_all A list of fitted Q-function objects for each regime, or an empty list if q_list is NULL.

---

gen_no_trt_resp *Generate sample data for a two-stage SMART where responders are not re-randomized*

Description

This function generates data for a two-stage Sequential Multiple Assignment Randomized Trial (SMART) where responders are not re-randomized in the second stage. The function allows for different value patterns and treatment assignments.

Usage

```
gen_no_trt_resp(n, s2 = 100, block_rep = 2, r2p = 0.5)
```

Arguments

n	A positive integer. Number of individuals to generate data for.
s2	A positive numeric value. Variance of the error term. Default is 100.
block_rep	A positive integer. Number of times to duplicate treatments per block in the permuted block randomization. Default is 2.
r2p	A numeric value between 0 and 1. Probability of being a responder in the second stage. Default is 0.5.

Value

A data frame with n rows and the following columns:

t1 Study day of enrollment.

t2 Study day of second-stage randomization.

t3 Study day of outcome assessment.

a1 First-stage treatment assignment (0 or 1).

r2 Response indicator at stage 2 (0 = non-responder, 1 = responder).

a2 Second-stage treatment assignment (0 or 1 for non-responders, 0 for responders).

x11 Baseline covariate (continuous, range 25–75).

x12 Baseline covariate (binary, 0 or 1).

x21 Stage 2 covariate (continuous, range 0–1).

y Continuous outcome (higher is better).

id Individual identifier (1 to n).

Examples

```
set.seed(1)
dat <- gen_no_trt_resp(n=400, s2=100, block_rep=2, r2p = 0.5)
head(dat)
```

---

get_an *Compute the An matrix for the sandwich variance estimator*

Description

Computes the An matrix, defined as $-\partial\Psi/\partial\theta$, where Ψ is the stacked estimating equations vector. The estimating equations are ordered to match Bn: $\pi_1, \ldots, \pi_K$, $\nu_1, \ldots, \nu_{K+1}$, $\beta_1^{(\ell)}, \ldots, \beta_K^{(\ell)}$ for each regime ℓ, and the value estimating equations $V_1, \ldots, V_L$. Based on Section 7 of Boos and Stefanski for the robust sandwich matrix.

Usage

```
get_an(
  df,
  pis,
  p_fits,
  nus,
  q_all,
  values,
  regime_all,
  dfs,
  feasible_sets_indicator,
  q_list
)
```

Arguments

df	A data frame containing the trial data, including treatment assignments, covariates, outcomes, and a kappa column.
pis	A data frame of estimated propensity scores with columns pi1, ..., piK, one column per stage.
p_fits	A list of fitted propensity score model objects (one per stage), each a modelObj fit object.
nus	A list as returned by get_nu, containing the estimated stage probabilities nu, the sample size ns, and nd.
q_all	A list of fitted outcome regression objects for each regime, as returned by estimate_values. Can be an empty list when using IPW estimation.
values	A numeric vector of estimated regime values.
regime_all	A list of regime objects, each containing a regime matrix and a regime_ind indicator matrix.
dfs	A list of data frames of value term components for each regime, as returned by estimate_values.
feasible_sets_indicator	A logical value indicating whether feasible sets are present in the trial design (i.e., some treatments are deterministic based on response status).
q_list	A list of outcome regression model specifications (one per stage), used to determine the number of Q-function models at each stage (e.g., separate models for responders and non-responders).

Value

A numeric matrix representing the An component of the sandwich variance estimator, with dimensions equal to the total number of estimated parameters.

---

get_bn *Compute the Bn matrix for the sandwich variance estimator*

Description

Computes the Bn matrix, defined as $n^{-1}\sum_{i=1}^{n}\Psi_i\Psi_i^T$, the empirical variance of the estimating equations. Each row of the individual-level estimating equation matrix corresponds to one subject, with columns ordered as $\pi_1,\ldots,\pi_K,\nu_1,\ldots,\nu_{K+1},\beta_1^{(\ell)},\ldots,\beta_K^{(\ell)}$ for each regime ℓ, and the value estimating equations $V_1,\ldots,V_L$. Based on Section 7 of Boos and Stefanski for the robust sandwich matrix.

Usage

```
get_bn(df, p_fits, nus, q_all, dfs, values, feasible_sets_indicator, q_list)
```

Arguments

df	A data frame containing the trial data, including treatment assignments, covariates, outcomes, and a kappa column.
p_fits	A list of fitted propensity score model objects (one per stage), each a modelObj fit object.
nus	A list as returned by get_nu, containing the estimated stage probabilities nu, the sample size ns, and nd.
q_all	A list of fitted outcome regression objects for each regime, as returned by estimate_values. Can be an empty list when using IPW estimation.
dfs	A list of data frames of value term components for each regime, as returned by estimate_values.
values	A numeric vector of estimated regime values.
feasible_sets_indicator	A logical value indicating whether feasible sets are present in the trial design (i.e., some treatments are deterministic based on response status).
q_list	A list of outcome regression model specifications (one per stage), used to determine the number of Q-function models at each stage (e.g., separate models for responders and non-responders).

Value

A numeric matrix representing the Bn component of the sandwich variance estimator, with dimensions equal to the total number of estimated parameters.

---

get_bounds *Compute all stopping boundaries for a two-analysis group sequential design*

Description

Wrapper function that computes the stopping boundaries for both the first and second analyses of a group sequential design with multiple treatment regimes. Calls get_first_bound and get_next_bound sequentially.

Usage

```
get_bounds(
  alpha = 0.05,
  inf_frac = c(0.5, 1),
  spend_fn = "OF",
  corr = diag(x = 1, nrow = 1, ncol = 1),
  test_type = "one-sided",
  lambda = 0.1,
  tol = 1e-06,
  max_iter = 1000
)
```

Arguments

alpha	A numeric value specifying the overall type I error rate to control. Default is 0.05.
inf_frac	A numeric vector of information fractions indicating when analyses are conducted.
spend_fn	A character string specifying the alpha spending function. Either "OF" (O'Brien-Fleming) or "Pocock".
corr	A correlation matrix of Z-statistics at analysis time s. Should have dimension $L \times L$.
test_type	A character string specifying the type of test to be performed. Either "one-sided" or "two-sided". Default is "one-sided".
lambda	A numeric value for the initial step size used in the iterative boundary search. Default is 0.1.
tol	A numeric value specifying the convergence tolerance. Default is 1e-6.
max_iter	A positive integer specifying the maximum number of iterations. Default is 1000.

Value

A list with the following components:

bounds / bound A numeric value (single analysis) or numeric vector (sequential) of stopping boundaries. The key is "bounds" for a single analysis and "bound" for multiple analyses.

spending A numeric value or vector of cumulative alpha spent at each analysis.

convergence A logical value or vector indicating whether the algorithm converged at each analysis.

iterations An integer value or vector of the number of iterations used at each analysis.

alpha The input type I error rate.

inf_frac The input information fractions.

spend_fn The input spending function name.

corr The input correlation matrix.

test_type The input test type.

---

get_bounds_chi *Compute the first chi-squared stopping boundary for a group sequential design*

Description

Determines the first analysis stopping boundary based on a chi-squared global test statistic. This is used when the comparison type is a global chi-squared test (testing whether any regime differs from the others) rather than individual regime comparisons against a fixed control.

Usage

```
get_bounds_chi(
  alpha = 0.05,
  inf_frac = c(0.5, 1),
  spend_fn = "OF",
  corr = diag(x = 1, nrow = 1, ncol = 1),
  mu = NULL,
  B = 1000001,
  seed = 1
)
```

Arguments

alpha	A numeric value specifying the overall type I error rate to control. Default is 0.05.
inf_frac	A numeric vector of information fractions indicating when analyses are conducted. Values should be between 0 and 1. The length determines the number of planned analyses S. Default is `c(0.5, 1)`.
spend_fn	A character string specifying the alpha spending function. Currently used to adjust the `iota` scaling factors for each analysis boundary. For `"OF"` (O'Brien-Fleming), set `iota` to the information fractions. For `"Pocock"`, use `iota = rep(1, S)`. Default is `"OF"`.
corr	A correlation matrix of the expected correlation between the Z-statistics of regimes against a fixed value. Should have dimension $L \times L$ by $L \times L$, where L is the number of regimes. For `inf_frac` with length greater than one, the correlation structure across multiple analyses is computed.
mu	An optional numeric vector of means for the multivariate normal distribution for all regimes at a single analysis used in the Monte Carlo simulation. Default is `NULL`, which uses a zero vector (null hypothesis).
B	A positive integer specifying the number of Monte Carlo samples. Default is 1000001.
seed	An integer seed for reproducibility of the Monte Carlo simulation. Default is 1.

Details

The function uses Monte Carlo simulation to estimate the boundary that controls the familywise type I error rate at a specified level. A contrast matrix C is constructed to form $L - 1$ linearly independent comparisons among L regimes, and the chi-squared statistic is computed as $(CZ)'(C\Sigma C')^{-1}(CZ)$.

Value

A list with the following components:

bound A numeric vector of chi-squared stopping boundaries, one per analysis.

spending A numeric vector of observed cumulative alpha spent at each analysis.

typeI The overall achieved type I error rate across all analyses.

convergence A logical value; always TRUE for Monte Carlo based computation.

iters The number of Monte Carlo samples used (B).

dfchi The degrees of freedom of the chi-squared statistic, equal to the rank of $C\Sigma C'$.

alpha The input type I error rate.

inf_frac The input information fractions.

spend_fn The input spending function name.

corr The input correlation matrix.

mu The input mean vector.

B The input number of Monte Carlo samples.

seed The input random seed.

`get_first_bound` *Compute the first stopping boundary for a group sequential design*

Description

Determines the first analysis stopping boundary that controls the familywise type I error rate at a specified level, adjusting for the multiplicity of multiple treatment regimes. Uses an iterative search to find the boundary value. This function works only for superiority (one-sided) testing.

Usage

```
get_first_bound(
  alpha = 0.05,
  inf_frac = c(0.5, 1),
  spend_fn = "OF",
  corr = diag(x = 1, nrow = 1, ncol = 1),
  test_type = "one-sided",
  lambda = 0.1,
  tol = 1e-06,
  max_iter = 1000
)
```

Arguments

alpha	A numeric value specifying the overall type I error rate to control. Default is 0.05.
inf_frac	A numeric vector of information fractions indicating when analyses are conducted. Values should be between 0 and 1.
spend_fn	A character string specifying the alpha spending function. Either "OF" (O'Brien-Fleming) or "Pocock".
corr	A correlation matrix of Z-statistics at the first time point, accounting for multiple regimes. Default is a 1x1 identity matrix.

test_type	A character string specifying the type of test to be performed. Either "one-sided" or "two-sided". Default is "one-sided".
lambda	A numeric value for the initial step size used in the iterative boundary search. Default is 0.1.
tol	A numeric value specifying the convergence tolerance for the type I error boundary. Default is 1e-6.
max_iter	A positive integer specifying the maximum number of iterations for the boundary search. Default is 1000.

Value

A list with the following components:

bound The computed stopping boundary for the first analysis.

typeI The achieved type I error rate at the boundary.

convergence A logical value indicating whether the algorithm converged.

iters The number of iterations used.

---

get_kappa *Compute the kappa (stage reached) for each individual*

Description

Determines the total number of stages each individual has reached by a specified analysis time t_s, based on their arrival times t1, ..., t_{K+1}.

Usage

```
get_kappa(df, t_s, K)
```

Arguments

df	A data frame containing columns t1, t2, ..., t_{K+1} representing the times at which each individual reaches each stage.
t_s	A numeric value specifying the analysis time point.
K	An integer specifying the number of treatment stages (decision points).

Value

An integer vector of length nrow(df) indicating the number of stages each individual has reached by time t_s. Values range from 0 (not yet enrolled) to $K + 1$ (outcome observed).

get_next_bound *Compute subsequent stopping boundaries for a group sequential design*

Description

Given the boundary from the first analysis, determines the stopping boundary for the s-th analysis that controls the overall type I error rate. Uses an iterative search with the joint distribution of test statistics across analyses.

Usage

```
get_next_bound(
  alpha = 0.05,
  inf_frac = c(0.5, 1),
  spend_fn = "OF",
  corr = diag(x = 1, nrow = 1, ncol = 1),
  test_type = "one-sided",
  lambda = 0.1,
  tol = 1e-06,
  prev_bound = NULL,
  s = 2,
  max_iter = 1000
)
```

Arguments

alpha	A numeric value specifying the overall type I error rate to control. Default is 0.05.
inf_frac	A numeric vector of information fractions indicating when analyses are conducted.
spend_fn	A character string specifying the alpha spending function. Either "OF" (O'Brien-Fleming) or "Pocock".
corr	A correlation matrix of Z-statistics across all analyses and regimes. Should have dimension $SL \times SL$ by $SL \times SL$.
test_type	A character string specifying the type of test to be performed. Either "one-sided" or "two-sided". Default is "one-sided".
lambda	A numeric value for the initial step size used in the iterative boundary search. Default is 0.1.
tol	A numeric value specifying the convergence tolerance. Default is 1e-6.
prev_bound	A numeric vector of previously computed boundaries from analyses $1, \ldots, s-1$, with dimension $(s-1) \times L$.
s	An integer indicating the current analysis number. Default is 2.
max_iter	A positive integer specifying the maximum number of iterations. Default is 1000.

Value

A list with the following components:

bound The computed stopping boundary for analysis `s`.

typeI The achieved cumulative type I error rate through analysis `s`.

convergence A logical value indicating whether the algorithm converged.

iters The number of iterations used.

`get_nu` *Estimate stage arrival probabilities (nu)*

Description

Estimates the probability that an individual has reached each stage given they are enrolled in the trial. Requires the data frame to have a `kappa` column indicating the stage reached. The returned list uses indexing such that `nu[[k]]` corresponds to stage k, and `nu[[K+1]]` is the probability that an individual has their final outcome observed.

Usage

```
get_nu(df, K)
```

Arguments

`df`	A data frame containing a `kappa` column indicating the stage reached by each individual.
`K`	An integer specifying the number of treatment stages (decision points).

Value

A list with the following components:

nu A list of length $K+1$ where `nu[[k]]` is the estimated probability that an individual has reached stage k given enrollment.

ns The total number of individuals enrolled in the trial (`kappa > 0`).

nd The proportion of individuals who have reached their last treatment stage and have their outcome observed.

get_q_coefs *Extract coefficients from all fitted Q-function models*

Description

Extracts and concatenates the regression coefficients from all fitted Q-function models across all regimes and stages.

Usage

```
get_q_coefs(q_all)
```

Arguments

q_all	A list of fitted outcome regression objects for each regime, as returned by estimate_values.

Value

A numeric vector of all Q-function regression coefficients, ordered by regime and then by stage.

get_q_fits *Fit outcome regression (Q-function) models across all stages for a single regime*

Description

Fits Q-function models backwards from the last stage to the first for a single treatment regime. At each stage, both regime-modified and unmodified predicted values are computed. Handles feasible sets (where responders may not be re-randomized) and interim analyses.

Usage

```
get_q_fits(df, q_list, regime, feasible_sets_indicator = FALSE)
```

Arguments

df	A data frame containing the trial data, including treatment assignments, covariates, outcomes (y), and a kappa column.
q_list	A list of outcome regression model specifications (one per stage). Each element is either a modelObj object or a named list with elements "r0" and "r1" for non-responders and responders, respectively.
regime	A matrix of treatment assignments under the regime being evaluated, with columns corresponding to stages.
feasible_sets_indicator	A logical value indicating whether feasible sets are present. Default is FALSE. If individuals are not re-randomized (i.e. for some stage a response status prevents individuals from receiving a random treatment), then feasible_sets_indicator should be set to TRUE.

Value

A list with the following components:

q_fits A list of fitted `modelObj` objects, one per stage. When feasible sets with multiple models are used, the element is a named list with `"r0"` and `"r1"`.

mod_regime_vhats A matrix of regime-modified predicted values with columns `q1, ..., q_{K+1}`.

unmod_regime_vhats A matrix of unmodified predicted values with columns `q1_nochange, ..., q_{K+1}_nochange`.

`get_sample_size` *Determine sample size for a group sequential SMART design*

Description

For specified operating characteristics, iteratively increases the sample size until the desired power is achieved. Assumes the information fraction remains unchanged as the sample size increases, which is reasonable for small changes or at the design stage.

Usage

```
get_sample_size(
  variances,
  beta,
  delta,
  bounds,
  n_init = 100,
  corr = bounds$corr,
  inf_frac = bounds$inf_frac,
  n_split = NULL
)
```

Arguments

`variances`	A numeric vector of length L of variances of the value estimators, i.e., $\sqrt{N} \times \mathrm{Cov}(\hat{\theta})$. These should reflect the population variances rather than sample variance or standard errors.
`beta`	A numeric value between 0 and 1 specifying the type II error rate. Power is `1 - beta`.
`delta`	A numeric vector of length L of differences between regime values and the null value (or control arm).
`bounds`	A numeric vector of length S with the stopping boundaries for analyses $1, \ldots, S$, or the boundaries from `get_bounds` function
`n_init`	A positive integer specifying the initial total trial sample size N to begin the search.

corr	A correlation matrix of dimension $L \times L$ between regime value estimators across all analyses.
inf_frac	A numeric vector of information fractions indicating when analyses are conducted.
n_split	A numeric vector indicating the proportion of the sample size at the analysis times s=1,...,S. If no argument is given, assumes the split is proportional to the information available.

Value

A list with the following components:

N A numeric vector of sample sizes at each analysis.

power The achieved power at the final sample size.

prop_rej A numeric vector of cumulative rejection probabilities at each analysis.

variances The input variances.

beta The input type II error rate.

delta The input alternative differences.

bounds The input stopping boundaries.

n_init The input initial sample size.

corr The input correlation matrix.

inf_frac The input information fractions.

n_split The input sample size split proportions.

---

get_sample_size_chi *Determine sample size for a chi-squared global test in a group sequential SMART design*

Description

For specified operating characteristics, iteratively increases the sample size until the desired power is achieved using a chi-squared global test statistic. The chi-squared test assesses whether any treatment regime differs from the others, rather than comparing individual regimes against a fixed control.

Usage

```
get_sample_size_chi(
  variances,
  beta,
  delta,
  bounds,
  n_init = 100,
  corr = bounds$corr,
```

```
  inf_frac = bounds$inf_frac,
  n_split = NULL,
  seed = 2,
  B = 10001,
  lambda = 20,
  max_iter = 100
)
```

Arguments

variances	A numeric vector of length L of variances of the value estimators, i.e., $\sqrt{N} \times \mathrm{Cov}(\hat{\theta})$. These should reflect the population variances rather than sample variance or standard errors.
beta	A numeric value between 0 and 1 specifying the type II error rate. Power is 1 - beta.
delta	A numeric vector of length L of change in the regime values under the alternative hypothesis.
bounds	A numeric vector of chi-squared stopping boundaries for analyses $1, \ldots, S$, or the output list from get_bounds_chi.
n_init	A positive integer specifying the initial total trial sample size N to begin the search. Default is 100.
corr	A correlation matrix of dimension $L \times L$ between regime value estimators. Defaults to the correlation from the bounds list if provided.
inf_frac	A numeric vector of information fractions indicating when analyses are conducted. Defaults to the information fractions from the bounds list if provided.
n_split	A numeric vector indicating the proportion of the sample size at the analysis times $s = 1, \ldots, S$. If NULL (default), assumes the split is proportional to the information fractions.
seed	An integer seed for reproducibility of the Monte Carlo simulation. Default is 2.
B	A positive integer specifying the number of Monte Carlo samples for power estimation. Default is 10001.
lambda	A positive numeric value specifying the incremental sample size increases when raising the sample size to find the required power via simulation. After this is found, iteration from $n - lambda$ to $n + lambda$ will be done to find the correct exact sample size.
max_iter	The maximum sample sizes n to evaluate.

Details

The function uses Monte Carlo simulation to estimate power at each candidate sample size. A contrast matrix C is constructed to form $L-1$ linearly independent comparisons among L regimes, and the chi-squared statistic is computed as $(CZ)'(C\Sigma C')^{-1}(CZ)$.

Value

A list with the following components:

N A numeric vector of sample sizes at each analysis.

power The achieved power at the final sample size.

prop_rej A numeric vector of cumulative rejection probabilities at each analysis.

variances The input variances.

beta The input type II error rate.

delta The input alternative differences.

bounds The input stopping boundaries.

n_init The input initial sample size.

corr The input correlation matrix.

inf_frac The input information fractions.

n_split The input sample size split proportions.

seed The input random seed.

B The input number of Monte Carlo samples.

lambda The input step size for the sample size search.

`iaipwe` *IAIPWE for K-stage SMARTs with up to 2 treatment options at each stage*

Description

The IAIPWE was developed for arbitrary K stage SMART and subsumes both IPWE and AIPWE. To implement the IPWE, q_list should be NULL and the t_s should be set to the maximum available time of the outcome observed. To implement the AIPWE, t_s should be set to the maximum available time of the outcome observed.

Usage

```
iaipwe(df, pi_list, q_list, regime_all, feasible_sets_indicator, t_s, B = NULL)
```

Arguments

`df`	A data frame containing the data. It should include columns for the treatment assignments, response status, covariates, and outcomes.
`pi_list`	A list of `modelObj` objects of length K specifying the propensity score model at each stage.
`q_list`	A list of `modelObj` objects of length K specifying the outcome regression model at each stage, or `NULL` for IPW-only estimation.

`regime_all` A list of length equal to the number of regimes to be estimated. Each element is a list with two components: `regime`, an n x K matrix of treatment assignments each individual would receive under that regime, and `regime_ind`, an n x K indicator matrix of whether each individual was consistent with that regime at each stage.

`feasible_sets_indicator` A logical value indicating whether feasible sets are present in the trial design. If `TRUE`, some treatments are assigned deterministically based on response status; if `FALSE`, all stages have random treatment assignment.

`t_s` A numeric value indicating the time point at which the analysis should occur.

`B` A positive integer specifying the number of empirical bootstrap samples to use for variance estimation, or `NULL` (default) to use the asymptotic sandwich variance estimator.

Details

If times are not recorded for the study, "dummy" times can be used with the analysis time set to be the maximum of the dummy times.

Value

A list with the following components:

values A numeric vector of estimated regime values.

se A numeric vector of standard errors for the regime values.

covariance The estimated L x L covariance matrix of the regime value estimators.

params A numeric vector of all estimated parameters.

nus A list of estimated stage arrival probabilities, as returned by `get_nu()`.

q_all A list of fitted Q-function objects for each regime.

regime_all The input regime list, returned for convenience.

dfs A list of value term matrices for each regime.

variance_choice A character string indicating the variance estimation method used (`"Asymptotic"` or `"Bootstrap"`).

chi_square A list with `Statistic` (the chi-squared test statistic for equality of regime values) and `p_value`.

`pcsttrial` *pcsttrial: Simulated Clinical Trial Data for Pain Coping Skills Training*

Description

A dataset containing simulated patient data to mimic the characteristics of the Pain Coping Skills Training trial presented in Manschot, Laber, and Davidian (2023).

Usage

```
pcsttrial
```

Format

A data frame with 284 rows and 14 variables:

pctchange Percent change in outcome from baseline at final assessment

height Patient height (cm)

weight Patient weight (kg)

comorbidity Comorbidity indicator (0 = no, 1 = yes)

painmed Pain medication use indicator (0 = no, 1 = yes)

chemo Chemotherapy indicator (0 = no, 1 = yes)

pctchangek2 Percent change in outcome at stage 2

adherence Treatment adherence measure, from 0.5 to 1 in increments of 0.1

a1 First-stage treatment assignment

a2 Second-stage treatment assignment

r2 Response indicator at stage 2 (0 = non-responder, 1 = responder)

study_day_enroll Study day of enrollment

study_day_rerand Study day of re-randomization

study_day_outcome Study day of outcome assessment

Details

Note: this dataset is for illustrative purposes and does not use the actual trial data.

---

pi_fits *Fit propensity score models for all stages*

Description

Fits propensity score models at each stage of the SMART, using only individuals who have reached that stage. Returns estimated propensity scores and fitted model objects for all stages.

Usage

```
pi_fits(df, p_list)
```

Arguments

`df`	A data frame containing the trial data, including treatment assignments (`a1`, `a2`, ...), covariates, and a `kappa` column.
`p_list`	A list of `modelObj` objects specifying the propensity score model at each stage.

Value

A list with the following components:

ps A data frame of estimated propensity scores with columns `pi1, ..., piK`. Individuals who have not reached a stage are assigned a value of 99.

p_fits A list of fitted `modelObj` objects, one per stage.

`pstep` *Fit a propensity score model for a single stage*

Description

Fits a propensity score model at a single stage of the SMART and returns the fitted probabilities. The probability that each individual received the treatment they actually received is computed.

Usage

```
pstep(pmodel, data, response, k)
```

Arguments

`pmodel`	A `modelObj` object specifying the propensity score model (typically a binomial GLM).
`data`	A data frame of individuals to fit the model on (typically those who have reached stage `k`).
`response`	A vector or data frame column of treatment assignments at stage `k`.
`k`	An integer indicating the stage number.

Value

A list with the following components:

pk A numeric vector of predicted probabilities that treatment is 1 at stage `k`.

ps A numeric vector of estimated propensity scores (probability that the individual received the treatment they actually received).

pfit The fitted `modelObj` object.

qstep *Fit an outcome regression (Q-function) model for a single stage*

Description

Fits a single Q-function model at one stage of the SMART. Predictions are obtained both for the unmodified data and for data where the treatment is set to the recommended regime.

Usage

```
qstep(qmodel, data, response, newdata, regime, txName)
```

Arguments

qmodel	A modelObj object specifying the outcome regression model.
data	A data frame of individuals used to fit the model (typically those who have completed the stage).
response	A numeric vector or single-column data frame of responses (pseudo-outcomes from later stages or observed outcomes).
newdata	A data frame of individuals for whom predictions are desired (typically those who have reached the stage).
regime	A vector of recommended treatment assignments under the regime being evaluated.
txName	A character string specifying the name of the treatment column to modify (e.g., "a1", "a2").

Value

A list with the following components:

hats_mod A numeric vector of predicted values under the regime-consistent treatment.

hats_unmod A numeric vector of predicted values under the actual (unmodified) treatment.

qfit The fitted modelObj object.

regime_list_no_trt_resp *Generate regime lists for a two-stage SMART where responders are not re-randomized*

Description

Constructs the treatment assignment matrices and consistency indicator matrices for each embedded regime in a two-stage SMART. For stages where response status determines treatment deterministically, the regime matrix is updated to reflect the responder treatment.

Usage

```
regime_list_no_trt_resp(
  emb_regimes,
  dat,
  resp_trt = list(r2 = list(0, 0, 0, 0))
)
```

Arguments

emb_regimes	A list of numeric vectors, each of length K, specifying the treatment codes (0 or 1) for non-responders at each stage. For example, `list(c(0,0), c(0,1), c(1,0), c(1,1))` encodes four regimes.
dat	A data frame containing the SMART data. Expected to contain columns `a1, ..., aK` (treatment assignments) and optionally `r1, ..., rK` (response indicators). Typically this is the output from `gen_no_trt_resp()`, but any data frame with the required columns may be used.
resp_trt	A named list where each element corresponds to a stage with deterministic responder treatment (e.g., `"r2"`). Each element is a list of length equal to the number of regimes, specifying the treatment assignment (0 or 1) for responders at that stage under each regime. Default is `list("r2" = list(0, 0, 0, 0))`.

Value

A list of length `length(emb_regimes)`. Each element is a list with two components:

regime An n x K matrix of treatment assignments each individual would receive if they followed that regime.

regime_ind An n x K indicator matrix (0 or 1) of whether each individual's observed treatment was consistent with the regime at each stage.

---

sim_treatment *Simulate treatment assignments for a single stage*

Description

Generates treatment assignments for `n` individuals given a specified number of possible treatments and a randomization probability function. This is a utility function for constructing SMART simulations with flexible randomization schemes. If an input data frame is provided, the treatment assignments are appended as a new column `a<stage>`.

Usage

```
sim_treatment(
  n = NULL,
  n_treatments,
  rand_prob_fn = function(n, n_treatments, prob) {
```

```
    sample(0:(n_treatments - 1), size
   = n, replace = TRUE, prob = prob)
},
 dat = NULL,
 stage = 1,
 randomize_response = NULL,
 prob = NULL
)
```

Arguments

n	A positive integer. Number of individuals to assign treatments to. If dat is provided, n is inferred from nrow(dat) and this argument is ignored.
n_treatments	A positive integer. Number of possible treatments. Treatments are coded as integers 0, 1, ..., n_treatments - 1.
rand_prob_fn	A function that takes n (the number of individuals), n_treatments (the number of treatment options), and prob (a numeric vector of probabilities) as arguments, and returns an integer vector of length n with treatment assignments coded as 0, 1, ..., n_treatments - 1. Default is Bernoulli randomization using the probabilities specified by prob.
dat	An optional data frame. If provided, the function appends the treatment assignments as a new column named a<stage> and returns the modified data frame. The data frame must not already contain a column with that name. Required when stage > 1.
stage	A positive integer indicating the stage number. The treatment column will be named paste0("a", stage). Default is 1. When stage > 1, dat must be provided and must contain columns a1, ..., a<stage-1>. Randomization is performed independently within each unique combination of prior treatments.
randomize_response	One of NULL (default), "Y", or "N". If "Y", the response indicator r<stage> is included in the grouping variable along with prior treatments a1, ..., a<stage-1> when randomizing. This allows different randomization within responder and non-responder subgroups. If "N", responders (r<stage> == 1) are assigned treatment 0 deterministically, and only non-responders are randomized within prior treatment groups. Both "Y" and "N" require stage > 1 and dat to contain a column named r<stage>. If NULL, response status is not used.
prob	A numeric vector or a named list specifying randomization probabilities. If a numeric vector, it is used as the probability weights for all groups (must have length equal to n_treatments). If a named list, the names should correspond to the group levels (formed by interaction() of prior treatment columns and optionally the response column), and each element should be a numeric vector of length n_treatments giving the group-specific probabilities. Default is NULL, which uses equal probability across treatments.

Value

If dat is NULL (default), an integer vector of length n with treatment assignments coded as 0, 1, ..., n_treatments - 1. If dat is provided, the input data frame with a new integer column

a<stage> appended.

Examples

```
# Default: Bernoulli randomization with 2 treatments (prob 0.5)
set.seed(1)
a <- sim_treatment(n = 100, n_treatments = 2)
table(a)

# Unequal randomization: 70% to treatment 0, 30% to treatment 1
set.seed(1)
a <- sim_treatment(n = 100, n_treatments = 2, prob = c(0.7, 0.3))
table(a)

# Three treatments with equal probability
set.seed(1)
a <- sim_treatment(n = 300, n_treatments = 3)
table(a)

# With an input data frame
set.seed(1)
df <- data.frame(x1 = rnorm(100), x2 = rbinom(100, 1, 0.5))
df <- sim_treatment(n_treatments = 2, dat = df)
head(df)

# With an input data frame at stage 2
set.seed(1)
df <- data.frame(x1 = rnorm(100), a1 = rbinom(100, 1, 0.5))
df <- sim_treatment(n_treatments = 2, dat = df, stage = 2)
head(df)


# Stage 2 with different randomization probabilities depending on a1:
# equal (50/50) if a1 == 0, unequal (70/30) if a1 == 1
# Using a named list for prob, where names match group levels
set.seed(1)
n <- 200
df <- data.frame(a1 = rbinom(n, 1, 0.5))
df <- sim_treatment(n_treatments = 2, dat = df, stage = 2,
                    prob = list("0" = c(0.5, 0.5), "1" = c(0.7, 0.3)))
table(df$a1, df$a2)
```

---

smart_design *Compute stopping boundaries and sample size for a group sequential SMART*

Description

Calculates the stopping boundaries and required sample size for a group sequential design with multiple treatment regimes embedded in a SMART. The boundaries are found first to control the

type I error rate, then the sample size is determined to achieve the desired power. This function wraps `get_bounds` and `get_sample_size`.

Usage

```
smart_design(
  test_type = "one-sided",
  comp_type = "fixed.control",
  alpha = 0.05,
  beta = 0.1,
  delta = NULL,
  n_init = 400,
  inf_frac = 1,
  spend_fn = "OF",
  corr = NULL,
  variances = NULL,
  lambdaB = 0.1,
  lambdaC = 20,
  tol = 1e-06,
  max_iterB = 1000,
  max_iterC = 100,
  seed = 1,
  Bb = 100001,
  Bc = 10001,
  mu = NULL
)
```

Arguments

`test_type`	A character string specifying the type of hypothesis test. A test type of either one-sided or two-sided. Two-sided testing must be symmetric. This input is ignored for a global Chi-squared test.
`comp_type`	A character string specifying what comparison will be tested. Character string for either "fixed.control" or "global.chi.sq".
`alpha`	A numeric value between 0 and 1 specifying the overall familywise type I error rate. Default is `0.05`.
`beta`	A numeric value between 0 and 1 specifying the type II error rate. Power is `1 - beta`. Default is `0.1` for power of 0.9.
`delta`	A numeric vector of length equal to the number of treatment regimes specifying the alternative differences between each regime's value and the null (or control) value.
`n_init`	A positive integer specifying the initial total sample size to begin the iterative sample size search.
`inf_frac`	A numeric vector of information fractions indicating when analyses are conducted. Values should be between 0 and 1, with the last element equal to 1. Default is `1` (single analysis). For example, `c(0.5, 1)` specifies an interim analysis at 50\% information and a final analysis.

`spend_fn` A character string specifying the alpha spending function. Either `"OF"` (O'Brien-Fleming, default) or `"Pocock"`.

`corr` A correlation matrix of Z-statistics across all regimes and analyses. The dimension should be $L \times k$ by $L \times k$, where L is the number of treatment regimes. This matrix accounts for both within-analysis correlations between regimes and across-analysis correlations due to shared participants.

`variances` A numeric vector of length equal to the number of treatment regimes giving the variance of each value estimator scaled by sample size, i.e., $N \times \mathrm{Var}(\hat{V}_d)$.

`lambdaB` A numeric value for the initial step size used in the iterative boundary search. Default is 0.1.

`lambdaC` A numeric value for the initial step size used in the iterative sample size search. Default is 20.

`tol` A numeric value specifying the convergence tolerance. Default is 1e-6.

`max_iterB` A positive integer specifying the maximum number of iterations for the boundary search. Default is 1000.

`max_iterC` A positive integer specifying the maximum number of iterations for the sample size search. Default is 100.

`seed` An integer seed for reproducibility of the Monte Carlo simulation. Default is 1.

`Bb` A positive integer specifying the number of Monte Carlo samples for the boundary computation. Default is 100001.

`Bc` A positive integer specifying the number of Monte Carlo samples for the sample size computation. Default is 10001.

`mu` A parameter vector for the multivariate normal distribution of the treatment effect mean, used only for the chi-square distribution in the case of a non-centrality assumption, though unlikely to be needed.

Value

A list with the following components:

boundaries The output from `get_bounds` (when `comp_type = "fixed.control"`) or `get_bounds_chi` (when `comp_type = "global.chi.sq"`), containing the stopping boundaries and associated metadata. See those functions for details.

sample.size The output from `get_sample_size` or `get_sample_size_chi`, containing the required sample sizes, achieved power, cumulative rejection probabilities, and echoed input parameters. If `delta` is `NULL` (no alternative specified), the sample size is not computed and this element is `NULL`.

inputs A list echoing all input parameters passed to `smart_design`.

`value_terms` *Compute the individual-level value terms for a single regime*

Description

Computes the $2K + 1$ coarsening-level value terms for each individual under a single treatment regime. These include augmentation terms for levels $r = 1, \ldots, 2K$ and the IPW term for $R = \infty$ (i.e., $R = 2K + 1$). When outcome regression models (`qs`) are provided, the augmented terms are computed; otherwise only the IPW term is non-zero.

Usage

```
value_terms(df, regime_ind, pis, qs, nus)
```

Arguments

`df`	A data frame containing the trial data, including treatment assignments, covariates, outcomes (`y`), and a `kappa` column.
`regime_ind`	A matrix of regime consistency indicators for a single regime, with columns indicating whether each individual followed the regime at each stage.
`pis`	A data frame of estimated propensity scores with columns `pi1, ..., piK`.
`qs`	A list containing outcome regression fitted values for a single regime (as returned by `get_q_fits`), or `NULL` for IPW estimation.
`nus`	A list as returned by `get_nu`, containing the estimated stage probabilities.

Value

A numeric matrix with `nrow(df)` rows and $2K + 1$ columns, where each column corresponds to a coarsening-level term in the value estimator.

Index