

Package: blockRAR (via r-universe)

June 29, 2026

Version 1.0.3

Title Block Design for Response-Adaptive Randomization

Description Computes power for response-adaptive randomization with a block design that captures both the time and treatment effect.

T. Chandereeng, R. Chappell (2019)

<[doi:10.48550/arXiv.1904.07758](https://doi.org/10.48550/arXiv.1904.07758)>.

Suggests testthat, rmarkdown, ggplot2, knitr

License MIT + file LICENSE

Encoding UTF-8

VignetteBuilder knitr

URL <https://github.com/thevaachandereng/blockRAR/>

BugReports <https://github.com/thevaachandereng/blockRAR/issues/>

Imports dplyr, magrittr, ldbounds, methods, arm

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Thevaa Chandereeng [aut, cre, cph] (ORCID:

<<https://orcid.org/0000-0003-4078-9176>>), Rick Chappell [aut, cph]

Maintainer Thevaa Chandereeng <thevaasiinen@gmail.com>

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

Date/Publication 2026-06-29 13:50:07 UTC

RemoteUrl <https://github.com/cran/blockRAR>

RemoteRef HEAD

RemoteSha 4e690781c2ea78e4174b656b833be059ac0ee891

Contents

bdpbinomial	2
binomialbayes	5
binomialfreq	7
prop_strata	8

bdpbinomial

*Bayesian Discount Prior: Binomial counts***Description**

bdpbinomial is used for estimating posterior samples from a binomial outcome where an informative prior is used. The prior weight is determined using a discount function. This code is modeled after the methodologies developed in Haddad et al. (2017).

Usage

```
bdpbinomial(
  y_t = NULL,
  N_t = NULL,
  y0_t = NULL,
  N0_t = NULL,
  y_c = NULL,
  N_c = NULL,
  y0_c = NULL,
  N0_c = NULL,
  discount_function = "identity",
  alpha_max = 1,
  fix_alpha = FALSE,
  a0 = 1,
  b0 = 1,
  number_mcmc = 10000,
  weibull_scale = 0.135,
  weibull_shape = 3,
  method = "mc",
  compare = TRUE
)
```

Arguments

y_t	scalar. Number of events for the current treatment group.
N_t	scalar. Sample size of the current treatment group.
y0_t	scalar. Number of events for the historical treatment group.
N0_t	scalar. Sample size of the historical treatment group.
y_c	scalar. Number of events for the current control group.
N_c	scalar. Sample size of the current control group.
y0_c	scalar. Number of events for the historical control group.
N0_c	scalar. Sample size of the historical control group.

discount_function	character. Specify the discount function to use. Currently supports weibull, scaledweibull, and identity. The discount function scaledweibull scales the output of the Weibull CDF to have a max value of 1. The identity discount function uses the posterior probability directly as the discount weight. Default value is "identity".
alpha_max	scalar. Maximum weight the discount function can apply. Default is 1. For a two-arm trial, users may specify a vector of two values where the first value is used to weight the historical treatment group and the second value is used to weight the historical control group.
fix_alpha	logical. Fix alpha at alpha_max? Default value is FALSE.
a0	scalar. Prior value for the beta rate. Default is 1.
b0	scalar. Prior value for the beta rate. Default is 1.
number_mcmc	scalar. Number of Monte Carlo simulations. Default is 10000.
weibull_scale	scalar. Scale parameter of the Weibull discount function used to compute alpha, the weight parameter of the historical data. Default value is 0.135. For a two-arm trial, users may specify a vector of two values where the first value is used to estimate the weight of the historical treatment group and the second value is used to estimate the weight of the historical control group. Not used when discount_function = "identity".
weibull_shape	scalar. Shape parameter of the Weibull discount function used to compute alpha, the weight parameter of the historical data. Default value is 3. For a two-arm trial, users may specify a vector of two values where the first value is used to estimate the weight of the historical treatment group and the second value is used to estimate the weight of the historical control group. Not used when discount_function = "identity".
method	character. Analysis method with respect to estimation of the weight parameter alpha. Default method "mc" estimates alpha for each Monte Carlo iteration. Alternate value "fixed" estimates alpha once and holds it fixed throughout the analysis. See the the bdpbinomial vignette vignette("bdpbinomial-vignette", package="bayesDP") for more details.
compare	logical. Should a comparison object be included in the fit? For a one-arm analysis, the comparison object is simply the posterior chain of the treatment group parameter. For a two-arm analysis, the comparison object is the posterior chain of the treatment effect that compares treatment and control. If compare=TRUE, the comparison object is accessible in the final slot, else the final slot is NULL. Default is TRUE.

Details

bdpbinomial uses a two-stage approach for determining the strength of historical data in estimation of a binomial count mean outcome. In the first stage, a *discount function* is used that defines the maximum strength of the historical data and discounts based on disagreement with the current data. Disagreement between current and historical data is determined by stochastically comparing the respective posterior distributions under noninformative priors. With binomial data, the comparison is the probability (p) that the current count is less than the historical count. The comparison metric

p is then input into the Weibull discount function and the final strength of the historical data is returned (α).

In the second stage, posterior estimation is performed where the discount function parameter, α , is used incorporated in all posterior estimation procedures.

To carry out a single arm (OPC) analysis, data for the current treatment (y_t and N_t) and historical treatment (y_{0_t} and N_{0_t}) must be input. The results are then based on the posterior distribution of the current data augmented by the historical data.

To carry our a two-arm (RCT) analysis, data for the current treatment and at least one of current or historical control data must be input. The results are then based on the posterior distribution of the difference between current treatment and control, augmented by available historical data.

For more details, see the `bdpbinomial` vignette:

```
vignette("bdpbinomial-vignette", package="bayesDP")
```

Value

`bdpbinomial` returns an object of class "bdpbinomial".

An object of class `bdpbinomial` is a list containing at least the following components:

`posterior_treatment` list. Entries contain values related to the treatment group:

`alpha_discount` numeric. Alpha value, the weighting parameter of the historical data.

`p_hat` numeric. The posterior probability of the stochastic comparison between the current and historical data.

`posterior` vector. A vector of length `number_mcmc` containing posterior Monte Carlo samples of the event rate of the treatment group. If historical treatment data is present, the posterior incorporates the weighted historical data.

`posterior_flat` vector. A vector of length `number_mcmc` containing Monte Carlo samples of the event rate of the current treatment group under a flat/non-informative prior, i.e., no incorporation of the historical data.

`prior` vector. If historical treatment data is present, a vector of length `number_mcmc` containing Monte Carlo samples of the event rate of the historical treatment group under a flat/non-informative prior.

`posterior_control` list. Similar entries as `posterior_treatment`. Only present if a control group is specified.

`final` list. Contains the final comparison object, dependent on the analysis type:

One-arm analysis: vector. Posterior chain of binomial proportion.

Two-arm analysis: vector. Posterior chain of binomial proportion difference comparing treatment and control groups.

`args1` list. Entries contain user inputs. In addition, the following elements are ouput:

`arm2` binary indicator. Used internally to indicate one-arm or two-arm analysis.

`intent` character. Denotes current/historical status of treatment and control groups.

References

Haddad, T., Himes, A., Thompson, L., Irony, T., Nair, R. MDIC Computer Modeling and Simulation working group.(2017) Incorporation of stochastic engineering models as prior information in Bayesian medical device trials. *Journal of Biopharmaceutical Statistics*, 1-15.

Examples

```
# One-arm trial (OPC) example
fit <- bdpbinomial(y_t = 10,
                  N_t = 500,
                  y0_t = 25,
                  N0_t = 250,
                  method = "fixed")

# Two-arm (RCT) example
fit2 <- bdpbinomial(y_t = 10,
                   N_t = 500,
                   y0_t = 25,
                   N0_t = 250,
                   y_c = 8,
                   N_c = 500,
                   y0_c = 20,
                   N0_c = 250,
                   method = "fixed")
```

binomialbayes

Block Design for Response-Adaptive Randomization for Binomial Data

Description

Simulation for binomial counts for block design for response-adaptive randomization with time as a confounding

Usage

```
binomialbayes(
  p_control,
  p_treatment,
  N_total,
  block_number = 4,
  drift = 0,
  simulation = 10000,
  a0 = 0.5,
  b0 = 0.5,
  p = "n/2N",
  number_mcmc = 10000,
  prob_accept_ha = 0.95,
  early_success_prob = 0.99,
  futility_prob = 0.01,
  alternative = "greater",
  size_equal_randomization = 20,
  min_patient_earlystop = 20,
```

```

    max_prob = 0.8
)

```

Arguments

<code>p_control</code>	scalar. Proportion of events under the control arm.
<code>p_treatment</code>	scalar. Proportion of events under the treatment arm.
<code>N_total</code>	scalar. Total sample size.
<code>block_number</code>	scalar. Number of blocks or time levels. The default is set to 4. If <code>block_number</code> is set to 1. This is a traditional RCT design.
<code>drift</code>	scalar. The increase or decrease in proportion of event over time. In this case, the proportion of failure changes in each block by the number of patient accrued over the total sample size. The full drift effect is seen in the final block.
<code>simulation</code>	scalar. Number of simulation to be ran. The default is set to 10000.
<code>a0</code>	scalar. Prior value for the beta rate $\text{Beta}(a0, b0)$. Default is 0.5.
<code>b0</code>	scalar. Prior value for the beta rate $\text{Beta}(a0, b0)$. Default is 0.5.
<code>p</code>	scalar. Power for randomization ratio.
<code>number_mcmc</code>	scalar. Number of Monte Carlo Markov Chain draws in sampling posterior.
<code>prob_accept_ha</code>	scalar. Probability of accepting alternative hypothesis.
<code>early_success_prob</code>	scalar. Probability of stopping early for success.
<code>futility_prob</code>	scalar. Probability of stopping early for futility.
<code>alternative</code>	character. A string specifying the alternative hypothesis, must be one of "less" or "greater" (default).
<code>size_equal_randomization</code>	scalar. The number of run in patients because adaptive randomization is applied.
<code>min_patient_earlystop</code>	scalar. Minimum number of patients before early stopping rule is applied.
<code>max_prob</code>	scalar. The maximum probability for assigning to treatment/control group is 0.8.

Value

	a list with details on the simulation.
<code>power</code>	scalar. The power of the trial, ie. the proportion of success over the number of simulation ran.
<code>p_control_estimate</code>	scalar. The estimated proportion of events under the control group.
<code>p_treatment_estimate</code>	scalar. The estimated proportion of events under the treatment group.
<code>N_enrolled</code>	vector. The number of patients enrolled in the trial (sum of control and experimental group for each simulation.)
<code>N_control</code>	vector. The number of patients enrolled in the control group for each simulation.
<code>N_control</code>	vector. The number of patients enrolled in the experimental group for each simulation.
<code>randomization_ratio</code>	matrix. The randomization ratio allocated for each block.

Examples

```
binomialbayes(p_control = 0.20, p_treatment = 0.30, N_total = 100, simulation = 3)
binomialbayes(p_control = 0.50, p_treatment = 0.30, N_total = 100, simulation = 3)
```

binomialfreq	<i>Block Design for Response-Adaptive Randomization for Binomial Data</i>
--------------	---

Description

Simulation for binomial counts for block design for response-adaptive randomization with time as a confounding

Usage

```
binomialfreq(
  p_control,
  p_treatment,
  N_total,
  block_number = 4,
  drift = 0,
  simulation = 10000,
  conf_int = 0.95,
  alternative = "greater",
  correct = FALSE,
  early_stop = FALSE,
  size_equal_randomization = 20,
  min_patient_earlystop = 20,
  max_prob = 0.8
)
```

Arguments

p_control	scalar. Proportion of events under the control arm.
p_treatment	scalar. Proportion of events under the treatment arm.
N_total	scalar. Total sample size.
block_number	scalar. Number of blocks or time levels. The default is set to 4. If block_number is set to 1. This is a traditional RCT design.
drift	scalar. The increase or decrease in proportion of event over time. In this case, the proportion of failure changes in each block by the number of patient accrued over the total sample size. The full drift effect is seen in the final block.
simulation	scalar. Number of simulation to be ran. The default is set to 10000.
conf_int	scalar. Confidence level of the interval.

alternative	character. A string specifying the alternative hypothesis, must be one of "less" or "greater" (default).
correct	logical. A logical indicating whether to apply continuity correction when computing the test statistic: one half is subtracted from all O - E differences; however, the correction will not be bigger than the differences themselves.
early_stop	logical. A logical indicating whether the trials are stopped early for success or futility.
size_equal_randomization	scalar. The number of run in patients because adaptive randomization is applied.
min_patient_earlystop	scalar. Minimum number of patients before early stopping rule is applied.
max_prob	scalar. The maximum probability for assigning to treatment/control group is 0.8.

Value

	a list with details on the simulation.
power	scalar. The power of the trial, ie. the proportion of success over the number of simulation ran.
N_enrolled	vector. The number of patients enrolled in the trial (sum of control and experimental group for each simulation.)
N_control	vector. The number of patients enrolled in the control group for each simulation.
N_control	vector. The number of patients enrolled in the experimental group for each simulation.
randomization_ratio	matrix. The randomization ratio allocated for each block.

Examples

```
binomialfreq(p_control = 0.7, p_treatment = 0.65, N_total = 200,
             block_number = 2, simulation = 3)
binomialfreq(p_control = 0.5, p_treatment = 0.40, N_total = 200,
             block_number = 2, simulation = 3, drift = -0.15)
```

prop_strata

Stratified Proportion Estimate for Binomial Data

Description

Computing the proportion of treatment difference for stratified data. The stratification is done over time.

Usage

```
prop_strata(treatment, outcome, block)
```

Arguments

treatment	vector. The vector with treatment assignment, 0 for control and 1 for treatment group.
outcome	vector. The vector with outcome, 0 for failure and 1 for success. Must be the same length as treatment variable.
block	vector. The vector with factor level of the block. Must be same length as treatment variable.

Value

the weighted mean of proportion difference (treatment - control).

Examples

```
set.seed(20999)
prop_strata(c(0, 1, 1, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0),
            c(0, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1),
            as.factor(rep(1:3, each = 5)))
```

Index

[bdpbinomial](#), [2](#)
[bdpbinomial](#), ANY-method ([bdpbinomial](#)), [2](#)
[binomialbayes](#), [5](#)
[binomialfreq](#), [7](#)
[prop_strata](#), [8](#)