

Package ‘blockRAR’

September 8, 2019

Version 1.0.1

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. Chandereng, R. Chappell (2019) <arXiv:1904.07758>.

Suggests testthat, rmarkdown, pkgdown, devtools, ggplot2, knitr

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

VignetteBuilder knitr

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

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

Imports dplyr, magrittr, ldbounds, tibble, bayesDP, arm

NeedsCompilation no

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

Maintainer Thevaa Chandereng <chandereng@wisc.edu>

Repository CRAN

Date/Publication 2019-09-08 07:10:02 UTC

R topics documented:

binomialbayes	2
binomialfreq	3
prop_strata	4
Index	6

binomialbayes	<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

```
binomialbayes(p_control, p_treatment, N_total, block_number = 4,
  drift = 0, simulation = 10000, a0 = 0.5, b0 = 0.5, p = 0.5,
  number_mcmc = 10000, prob_accept_ha = 0.95,
  early_success_prob = 0.99, futility_prob = 0.01,
  alternative = "greater")
```

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.
a0	scalar. Prior value for the beta rate Beta(a0, b0). Default is 0.5.
b0	scalar. Prior value for the beta rate Beta(a0, b0). Default is 0.5.
p	scalar. Power for randomization ratio.
number_mcmc	scalar. Number of Monte Carlo Markov Chain draws in sampling posterior.
prob_accept_ha	scalar. Probability of accepting alternative hypothesis.
early_success_prob	scalar. Probability of stopping early for success.
futility_prob	scalar. Probability of stopping early for futility.
alternative	character. A string specifying the alternative hypothesis, must be one of "less" or "greater" (default).

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.

`p_control_estimate` scalar. The estimated proportion of events under the control group.

`p_treatment_estimate` scalar. The estimated proportion of events under the treatment group.

`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.

Examples

```
binomialbayes(p_control = 0.20, p_treatment = 0.30, N_total = 100, simulation = 10)
```

```
binomialbayes(p_control = 0.50, p_treatment = 0.30, N_total = 100, simulation = 5)
```

 binomialfreq

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

```
binomialfreq(p_control, p_treatment, N_total, block_number = 4,
  drift = 0, simulation = 10000, conf_int = 0.95,
  alternative = "greater", correct = FALSE, replace = TRUE,
  early_stop = FALSE)
```

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 IO - EI differences; however, the correction will not be bigger than the differences themselves.
replace	logical. should sampling be with replacement? If replace is set to FALSE (default), the 0 for control, 1 for treatment is replicated to the closest integer and this vector is sampled with no replacement. If replace is set to TRUE, the sampling is done based on randomization ratio provided with replacement.
early_stop	logical. A logical indicating whether the trials are stopped early for success or futility.

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.

p_control_estimate scalar. The estimated proportion of events under the control group.

p_treatment_estimate scalar. The estimated proportion of events under the treatment group.

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.

Examples

```
binomialfreq(p_control = 0.7, p_treatment = 0.65, N_total = 200,
             block_number = 2, simulation = 100)
binomialfreq(p_control = 0.5, p_treatment = 0.40, N_total = 200,
             block_number = 2, simulation = 100, 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

`binomialbayes`, 2

`binomialfreq`, 3

`prop_strata`, 4