Package ‘mtdesign’

July 23, 2025

Type Package

Title Mander and Thompson Designs

Version 0.1.2

Date 2024-11-01

Description Implements Mander & Thompson's (2010) <doi:10.1016/j.cct.2010.07.008> methods for two-stage designs optimal under the alternative hypothesis for phase II [cancer] trials. Also provides an implementation of Simon's (1989) <doi:10.1016/0197-2456(89)90015-9> original methodology and allows exploration of the operating characteristics of sub-optimal designs.

Language en-GB

License GPL (>= 3)

Encoding UTF-8

URL `https://github.com/openpharma/mtdesign`

BugReports `https://github.com/openpharma/mtdesign/issues`

RoxygenNote 7.3.2

Imports dplyr, futile.logger, ggplot2, magrittr, methods, Rcpp, rlang, tibble, tidyr

LinkingTo Rcpp, BH

Suggests parallel, testthat (>= 3.0.0), withr

Config/testthat/edition 3

NeedsCompilation yes

Author John Kirkpatrick [aut, cre] (ORCID: <`https://orcid.org/0000-0001-7790-3708`>)

Maintainer John Kirkpatrick <`mtdesign@puzzledface.net`>

Repository CRAN

Date/Publication 2024-11-05 14:10:15 UTC

Contents

`augmentGrid` *Augment a grid of candidate designs with type 1 and type 2 error probabilities, expected sample sizes and probabilities of early termination*

Description

Augment a grid of candidate designs with type 1 and type 2 error probabilities, expected sample sizes and probabilities of early termination

Usage

```
augmentGrid(d, parallel = TRUE, cores = NA, minChunkSize = 1e+05)
```

Arguments

`d`	a tibble created by ‘createGrid‘
`parallel`	use parallelisation if available
`cores`	the number of cores to use when parallelising. If <code>NA</code>, all available cores are requested
`minChunkSize`	The minimum size of the grid before paralellisation is attempted

Value

an augmented grid tibble

Usage Notes

Regardless of the value of ‘parallel‘, parallelisation is only used if the size of the grid is greater than <code>chunkSize</code>. If paralellisation is requested and needed, an exception is thrown if the parallel package is not available.

Examples

```
x <- createGrid(p0 = 0.1, p1 = 0.30, alpha = 0.1, beta = 0.1, nMin = 24, nMax = 32) %>%
  augmentGrid(parallel = FALSE)
```

`createGrid` *Create a grid of candidate designs*

Description

Create a grid of candidate designs

Usage

```
createGrid(
  p0,
  p1,
  alpha = 0.1,
  beta = NA,
  power = ifelse(is.na(beta), 0.9, 1 - beta),
  nMin = NA,
  nMax = NA,
  mander = TRUE
)
```

Arguments

`p0`	the response rate under the null hypothesis
`p1`	the response rate under the alternate hypothesis
`alpha`	the desired (one-sided) type 1 error rate
`beta`	the desired type 2 error rate
`power`	an alternative to `beta`
`nMin`	the lower bound for the search grid. If `NA`, `searchBounds` is called to provide an appropriate value
`nMax`	the lower bound for the search grid. If `NA`, `searchBounds` is called to provide an appropriate value
`mander`	is a Mander & Thompson or a Simon's design required?

Value

a tibble. See Usage notes for a list and description of columns.

Examples

```
# Standard use for a Simon's 2-stage design
x <- createGrid(p0 = 0.1, p1 = 0.5, alpha = 0.1, beta = 0.1, mander = FALSE)
# Custom search bounds for a Mander & Thompson design
y <- createGrid(p0 = 0.1, p1 = 0.4, alpha = 0.1, beta = 0.1, nMin = 20, nMax = 30)
```

`obtainDesign` *Finds optimal and minimax designs for either Simon's 2-stage or Mander & Thompson studies*

Description

`obtainDesign` is essentially a wrapper for calls to `createGrid` and `augmentGrid` followed by some simple filtering of the candidate designs to identify the optimal and minimax designs.

Usage

```
obtainDesign(
  grid = NULL,
  p0 = NA,
  p1 = NA,
  alpha = ifelse(is.null(grid), 0.05, NA),
  beta = ifelse(is.null(grid), 0.1, NA),
  fullGrid = FALSE,
  ...
)
```

Arguments

`grid`	Optional. A tibble created by `createGrid`. If `NULL`, then `p0`, `p1`, `alpha` and `beta` must be specified and `createGrid` is called to generate the required grid. If not `NULL` then `p0`, `p1`, `alpha` and `beta` are ignored
`p0`	the response rate under the null hypothesis
`p1`	the response rate under the alternate hypothesis
`alpha`	the desired (one-sided) type 1 error rate
`beta`	the desired type 2 error rate
`fullGrid`	should the full grid of all possible designs be returned, or simply the optimal and minimax solutions? For a Mander and Thompson design, optimal and minimax designs are returned for both the null and alternate hypotheses. See Usage Notes below.
`...`	passed to 'createGrid' or 'augmentGrid'. In particular `mander=TRUE` for a Mander & Thompson design or `mander=FALSE` for a Simon's 2-stage design.

Value

a tibble created by `createGrid`. If `fullGrid == FALSE` the table contains an additional column, `Criterion` indicating the type of design. Possible values for `Criterion` are "optimal" and "minimax" for Simon's designs and "optimalNull", "optimalAlt", "minimaxNull" and "minimaxAlt" for Mander & Thompson designs.

Usage notes

If `grid` is not `NULL` it is possible that none of the candidate designs are acceptable (that is, satisfy both the significance level and power requirements). If this is the case and `fullGrid == FALSE`, then an empty tibble is returned. If `versbose == TRUE` a warning message is also printed. If `fullGrid == TRUE` the full grid of all designs considered is returned. This can then be further interrogated to find optimal designs under constraints - for example with fixed stage sizes.

Examples

```
# Standard use (Simon's 2-stage design)
createGrid(p0 = 0.05, p1 = 0.25, alpha = 0.05, beta = 0.2, mander = FALSE) %>%
  augmentGrid(parallel = FALSE) %>%
  obtainDesign()
# Constrained stage sizes
createGrid(p0 = 0.25, p1 = 0.45, alpha = 0.05, beta = 0.2) %>%
  dplyr::filter(nStage1 == 8) %>%
  augmentGrid(parallel = FALSE) %>%
  obtainDesign()
```

`powerPlot` *Plot the power curve(s) for the given design(s)*

Description

Plot the power curve(s) for the given design(s)

Usage

```
powerPlot(grid, probs = seq(0, 1, 0.01))
```

Arguments

`grid`	the tibble containing the designs to be plotted
`probs`	the response rates for which the rejection probabilities are to be plotted

Value

the ggplot object containing the power curve(s)

Examples

```
createGrid(p0 = 0.05, p1 = 0.25, alpha = 0.05, beta = 0.2, mander = FALSE) %>%
  augmentGrid(cores = 2) %>%
  obtainDesign() %>%
  powerPlot(probs = seq(0, 0.5, 0.025))
```

`searchBounds` *Obtain default bounds for the construction of the search grid.*

Description

The formula used is the continuity corrected Normal approximation from Fleiss et al (2003).

Usage

```
searchBounds(p0, p1, alpha = 0.05, beta = 0.2, twoSided = TRUE)
```

Arguments

`p0`	the response rate under the null hypothesis
`p1`	the response rate under the alternate hypothesis
`alpha`	the desired (one-sided) type 1 error rate
`beta`	the desired type 2 error rate
`twoSided`	two- or one-sided significance level?

Value

a list with three elements: "n" - the single stage sample size from Fleiss et al; "min" - the lower bound, 0.8*n; "max" - the upper bound, 2*n. `floor()` and `ceiling()` are applied as appropriate.

Index