Package ‘VisualizeSimon2Stage’

April 23, 2024

Type Package

Title Visualize Simon's Two-Stage Design

Version 0.1.4

Date 2024-04-22

Description To visualize the probabilities of early termination, fail and success of Simon's two-stage design. To evaluate and visualize the operating characteristics of Simon's two-stage design.

License GPL-2

Imports methods

Encoding UTF-8

Language en-US

Depends R (>= 4.3.0), ggplot2

Suggests clinfun

RoxygenNote 7.3.1

Collate '0PACKAGE.R' 'Simon_pr.R' 'Simon_oc.R' 'autoplot.ph2simon.R' 'ph2simon_etc.R' 'r_simon.R'

NeedsCompilation no

Author Tingting Zhan [aut, cre, cph] (<https://orcid.org/0000-0001-9971-4844>)

Maintainer Tingting Zhan <tingtingzhan@gmail.com>

Repository CRAN

Date/Publication 2024-04-23 03:50:02 UTC

R topics documented:

`VisualizeSimon2Stage-package` *Visualize Simon\'s Two-Stage Design*

Description

Functions for visualizing the probabilities of early termination, fail and success of Simon's two-stage design. Functions for evaluating and visualizing the operating characteristics of Simon's two-stage design.

Author(s)

Maintainer: Tingting Zhan <`tingtingzhan@gmail.com`> (ORCID) [copyright holder]

References

doi:10.1016/01972456(89)900159

`https://www.ncss.com/software/pass/`

`autoplot.ph2simon` *Plot Simon's Two-Stage Design*

Description

Plot ph2simon object using **ggplot2**.

Usage

```
## S3 method for class 'ph2simon'
autoplot(object, ...)

## S3 method for class 'ph2simon'
autolayer(
  object,
  type = c("minimax", "optimal", "n1", "maximax"),
  n1 = stop("must provide `n1`"),
  n = stop("must provide `n`"),
  r1 = stop("must provide `r1`"),
  r = stop("must provide `r`"),
  pu = stop("must provide `pu`"),
  pa = stop("must provide `pa`"),
  ...
)
```

Arguments

`object`	ph2simon object
`...`	potential parameters, currently not in use
`type`	character scalar, one of `'minimax'`, `'optimal'`, `'n1'` and `'maximax'`
`n1, n`	(optional) integer scalars, Stage-1 sample size n_1 and total sample size n. Overridden if `object` is given
`r1, r`	(optional) integer scalars, number of response in Stage-1 r_1 and overall r required *exclusively*, i.e., passing Stage-1 means observing $> r_1$ response. Overridden if `object` is given
`pu, pa`	double scalars, see function ph2simon

Value

Function autoplot.ph2simon returns a ggplot object.

Function autolayer.ph2simon returns a list of ggproto and labels.

Examples

```
library(clinfun)
(x = ph2simon(pu = .2, pa = .4, ep1 = .05, ep2 = .1))
class(x)
autoplot(x, type = 'minimax')
autoplot(x, type = 'optimal')
autoplot(x, type = 'n1')
autoplot(x, type = 'maximax')

# example with r1 = 0
(des = ph2simon(pu = .05, pa = .3, ep1 = .05, ep2 = .2))
autoplot(des, type = 'optimal')
autoplot(des, type = 'minimax')
```

`print_ph2simon` *Alternate Print Method for a Simon's Two-Stage Design*

Description

An alternate print method for ph2simon object.

Usage

```
print_ph2simon(x, ...)
```

Arguments

`x`	a ph2simon object
`...`	additional parameters, currently not in use

Value

Function print_ph2simon does not have a returned value.

Note

We do not overwrite `clinfun:::print.ph2simon`.

Examples

```
library(clinfun)
(x = ph2simon(pu = .2, pa = .4, ep1 = .05, ep2 = .1))
print_ph2simon(x)
```

---

r_simon *Random Generator based on Simon's Two-Stage Design*

Description

Random generator based on Simon's two-stage design.

Usage

```
r_simon(R, n1, n, r1, prob)
```

Arguments

`R`	positive integer scalar, number of trials R
`n1, n`	positive integer scalars, Stage-1 sample size n_1 and total sample size n
`r1`	non-negative integer scalar, number of response in Stage-1 r_1 required *exclusively*, i.e., passing Stage-1 indicates observing $> r_1$ responses
`prob`	double scalar, true response rate p

Details

Function r_simon generates R copies of the number of responses y in the Simon's two-stage design. The conclusion of the trials are,

$y \leq r_1$ indicates early termination

$r_1 < y \leq r$ indicates failure to reject H_0

$y > r$ indicates success to reject H_0

Here r is not needed to *generate* the random number of responses y. Instead, r is needed to *determine* if the trial is a failure or a success. Therefore, r is not a parameter in r_simon.

Value

Function r_simon returns an integer vector of length R, which are the R copies of the number of responses in the Simon's two-stage design.

Examples

```
library(clinfun)
ph2simon(pu = .2, pa = .4, ep1 = .05, ep2 = .1) # using 'Optimal'
# set.seed if needed
(ys = r_simon(R = 10L, n1 = 19L, n = 54L, r1 = 4L, prob = .3))
table(cut.default(ys, breaks = c(0, 4L, 15L, 54L), right = TRUE,
  labels = c('early-termination', 'fail', 'success')))
```

Simon_oc *Simon_oc: Operating Characteristics of Simon's Two-Stage Design*

Description

Operating characteristics of Simon's two-stage design.

Usage

```
Simon_oc(
  prob,
  simon,
  type = c("minimax", "optimal", "n1", "maximax"),
  R = 10000L,
  n1 = stop("must provide `n1`"),
  n = stop("must provide `n`"),
  r1 = stop("must provide `r1`"),
  r = stop("must provide `r`"),
  ...
)
```

Arguments

`prob` *named* double vector, true response rate(s) p of (multiple) drug(s). The `names(prob)` should be the respective keyword(s) for the drug(s).

`simon` ph2simon object

`type` character scalar, type of Simon's two-stage design. Currently supports `'minimax'` (default) for minimum total sample size, `'optimal'` for minimum expected total sample size *under* p_0, `'n1'` for minimum Stage-1 sample size n_1, `'maximax'` to use up the user-provided maximum total sample size (parameter `nmax` of function ph2simon)

`R` integer scalar, number of simulations. Default `1e4L`.

`n1, n` (optional) integer scalars, Stage-1 sample size n_1 and total sample size n. Overridden if `simon` is given

`r1, r` (optional) integer scalars, number of response in Stage-1 r_1 and overall r required *exclusively*, i.e., passing Stage-1 means observing $> r_1$ response. Overridden if `simon` is given

`...` potential parameters, currently not in use

Details

..

Value

Function Simon_oc returns Simon_oc object

Slots

`maxResp` integer vector of same length as p, the frequencies of each regime having maximum response. The summation of `maxResp` is the number of simulation copies.

`Simon_maxResp` integer vector of same length as p, the frequencies of each regime having maximum response and success in Simon's two-stage trial.

Examples

```
library(clinfun)
(x = ph2simon(pu = .2, pa = .4, ep1 = .05, ep2 = .1))
Simon_oc(prob = c(A = .3, B = .2, C = .15), simon = x, type = 'minimax', R = 1e3L)
Simon_oc(prob = c(A = .3, B = .2, C = .15), simon = x, type = 'optimal', R = 1e3L)
```

`Simon_pr` *Simon_pr: Probabilities of a Simon's Two-Stage Design*

Description

Probability of frail (i.e., early termination), fail (to reject the null) and success (to reject the null) of a Simon's two-stage design, at given true response rate(s).

Usage

```
Simon_pr(prob, n1, n, r1, r)
```

Arguments

`prob` double vector, true response rate(s) p

`n1, n` positive integer scalars, Stage-1 sample size n_1 and total sample size n

`r1, r` non-negative integer scalars, number of response in Stage-1 r_1 and overall r required *exclusively*, i.e., passing Stage-1 indicates observing $> r_1$ responses, and rejecting H_0 indicates observing $> r$ responses.

Details

Given the Simon's two-stage design (n_1, r_1, n, r), for a response rate p, we have the number of Stage-1 positive responses $X_1 \sim \text{Binom}(n_1, p)$ and the number of Stage-2 positive responses $X_2 \sim \text{Binom}(n - n_1, p)$. Obviously X_1 and X_2 are independent.

The probability of early termination is $\text{Pr}(X_1 \leq r_1)$.

The probability of failure to reject H_0 is

$$\sum_{s_1=r_1+1}^{n_1} \text{Pr}(X_1 = s_1) \cdot \text{Pr}(X_2 \leq (r - s_1))$$

The probability of rejecting H_0 is

$$\sum_{s_1=r_1+1}^{n_1} \text{Pr}(X_1 = s_1) \cdot \text{Pr}(X_2 > (r - s_1))$$

Parameters nomenclature of `n1`, `n`, `r1` and `r` follows that of PASS and function ph2simon.

Value

Simon_pr returns Simon_pr object.

Slots

`.Data` ncol-3 double matrix, probability of frail (i.e., early termination), fail (to reject the null) and success (to reject the null), at each response rate p given in `@prob`

`eN` numeric vector, expected sample size(s) $\text{E}(N)$ for each of response rate(s) p

`prob` double vector, response rate(s) p

Examples

```
Simon_pr(prob = c(.2, .4), n1 = 15L, r1 = 3L, n = 24L, r = 7L)
```

`summary.ph2simon` *Summarize a Simon's Two-Stage Design*

Description

Summarize a Simon's two-stage design

Usage

```
## S3 method for class 'ph2simon'
summary(object, ...)
```

Arguments

`object`	ph2simon object
`...`	potential parameters, currently not in use

Value

Function summary.ph2simon returns a list with three (3) elements

- `'design'` integer matrix
- `'EN'` double matrix
- `'p'` double matrix

Examples

```
library(clinfun)
(x = ph2simon(pu = .2, pa = .4, ep1 = .05, ep2 = .1))
summary(x)
```

Index