Package 'VisualizeSimon2Stage'

March 18, 2025

Type Package

Title Visualize Simon's Two-Stage Design

Version 0.1.7

Date 2025-03-18

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 (>= 4.4.0), ggplot2

Suggests clinfun

RoxygenNote 7.3.2

NeedsCompilation no

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

Maintainer Tingting Zhan <tingtingzhan@gmail.com>

Repository CRAN

Date/Publication 2025-03-18 19:00:02 UTC

Contents

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)

References

doi:10.1016/01972456(89)900159

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

Examples

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

# an alternative print
print_ph2simon(x)

# language for a report
Sprintf.ph2simon(x, type = 'minimax')
Sprintf.ph2simon(x, type = 'optimal')
Sprintf.ph2simon(x, type = 'n1')
Sprintf.ph2simon(x, type = 'maximax')

autoplot(x, type = 'minimax')
autoplot(x, type = 'optimal')
autoplot(x, type = 'n1')
autoplot(x, type = 'maximax')

# operating characteristics
simon_oc(prob = c(A = .3, B = .2, C = .15), object = x, type = 'minimax')
simon_oc(prob = c(A = .3, B = .2, C = .15), object = x, type = 'optimal')

# example with r1 = 0
(x1 = clinfun::ph2simon(pu = .05, pa = .3, ep1 = .05, ep2 = .2))
# works with all of our functions
autoplot(x1, type = 'optimal') # etc.
```

`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, prob, object, ...)

## S3 method for class 'ph2simon'
r_simon(R, prob, object, ...)

## S3 method for class 'ph2simon4'
r_simon(
  R,
  prob,
  object,
  ...,
  r1 = object@r1,
  n1 = object@n1,
  r = object@r,
  n = object@n
)
```

Arguments

`R`	positive integer scalar, number of trials R
`prob`	double scalar, true response rate p
`object`	a ph2simon or ph2simon4 object
`...`	parameters of function `ph2simon4()`, most importantly `type`
`r1`, `n1`, `r`, `n`	(optional) integer scalars, see ph2simon4.

Details

Function `r_simon()` generates R copies of the number of responses y in **one** 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 of function `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

```
(x = clinfun::ph2simon(pu = .2, pa = .4, ep1 = .05, ep2 = .1))
set.seed(1532); r = r_simon(R = 1e2L, prob = .2, object = x)
set.seed(1532); r1 = r_simon.ph2simon4(R = 1e2L, prob = .2, r1 = 5L, n1 = 24L, r = 13L, n = 45L)
stopifnot(identical(r, r1))
table(attr(r, 'dx')) # look at beta, <10%
set.seed(24315); r2 = r_simon(R = 1e2L, prob = .4, object = x)
table(attr(r2, 'dx')) # look at alpha, <5%
```

---

simon_oc *Operating Characteristics of Simon's Two-Stage Design*

Description

Operating characteristics of **one** Simon's two-stage design.

Usage

```
simon_oc(prob, R, object, ...)

## S3 method for class 'ph2simon'
simon_oc(prob, R = 10000L, object, ...)

## S3 method for class 'ph2simon4'
simon_oc(
  prob,
  R = 10000L,
  object,
  ...,
  r1 = object@r1,
  n1 = object@n1,
  r = object@r,
  n = object@n
)
```

Arguments

`prob`	*named* double vector, true response rate(s) p of (multiple) drug(s). The `names(prob)` should be the name(s) of the drug(s).
`R`	integer scalar, number of simulations. Default `1e4L`.
`object`	ph2simon or ph2simon4 object
`...`	parameters of function `ph2simon4()`, most importantly `type`
`r1, n1, r, n`	(optional) integer scalars, see ph2simon4.

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.

---

simon_pr *Probabilities of one Simon's Two-Stage Design*

Description

Probabilities of frail (i.e., early termination) and success (to reject H_0) of **one** Simon's two-stage design, at given true response rate(s).

Usage

```
simon_pr(prob, object, ...)

## S3 method for class 'ph2simon'
simon_pr(prob, object, ...)

## S3 method for class 'ph2simon4'
simon_pr(
  prob,
  object,
  r1 = object@r1,
  n1 = object@n1,
  r = object@r,
  n = object@n,
  ...
)
```

Arguments

prob	double scalar or vector, true response rate(s) p
object	a ph2simon or ph2simon4 object
...	parameters of function ph2simon4(), most importantly type
r1, n1, r, n	(optional) integer scalars, see ph2simon4.

Details

Given one Simon's two-stage design (r_1, n_1, r, n) and a true 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

$$p_{\text{frail}} = \Pr(X_1 \leq r_1)$$

The probability of failure to reject H_0 is

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

The probability of successfully rejecting H_0 is

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

The expected sample size is

$$\text{E}(n) = p_{\text{frail}} \cdot n_1 + (1 - p_{\text{frail}}) \cdot n$$

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

Value

Function `simon_pr()` returns simon_pr object.

Slots

`frail` numeric scalar or vector, probabilities of frail (i.e., early termination) at given true response rate(s) p.

`reject` numeric scalar or vector, probabilities of success (to reject H_0) at given true response rate(s) p.

`eN` numeric scalar or vector, expected sample size(s) $\text{E}(n)$ at given true response rate(s) p.

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

Examples

```
(x = clinfun::ph2simon(pu = .2, pa = .4, ep1 = .05, ep2 = .1))
simon_pr(prob = c(.2, .3, .4), object = x)
simon_pr.ph2simon4(prob = c(.2, .3, .4), r1 = 5L, n1 = 24L, r = 13L, n = 45L) # internal use
```

Index