

# Package ‘VisualizeSimon2Stage’

October 19, 2023

**Type** Package

**Title** Visualize Simon's Two-Stage Design

**Version** 0.1.2

**Date** 2023-10-19

**Author** Tingting Zhan [aut, cre, cph]

**Maintainer** Tingting Zhan <tingtingzhan@gmail.com>

**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** 7.2.3

**Collate** 'Simon\_pr.R' 'Simon\_oc.R' 'VisualizeSimon2Stage\_info.R'  
'ph2simon\_etc.R'

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2023-10-19 15:40:04 UTC

## R topics documented:

autoplot.ph2simon . . . . .	2
print_ph2simon . . . . .	3
r_simon . . . . .	3
show,Simon_oc-method . . . . .	4
show,Simon_pr-method . . . . .	5

Simon_oc . . . . .	5
Simon_oc-class . . . . .	6
Simon_pr . . . . .	7
Simon_pr-class . . . . .	8
Sprintf.ph2simon . . . . .	8
Sprintf.Simon_oc . . . . .	9
summary.ph2simon . . . . .	9

<b>Index</b>	<b>11</b>
--------------	-----------

---

autoplot.ph2simon	<i>Plot Simon's Two-Stage Design</i>
-------------------	--------------------------------------

---

## Description

Plot `ph2simon` object using **ggplot2**.

## Usage

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

## Arguments

object	<code>ph2simon</code> object
...	potential parameters, currently not in use

## Value

Function `autoplot.ph2simon()` returns a `ggplot` object.

## 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	<i>Alternate Print Method for a Simon's Two-Stage Design</i>
----------------	--

---

**Description**

An alternate [print](#) method for [ph2simon](#) object.

**Usage**

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

**Arguments**

x	a <a href="#">ph2simon</a> 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	<i>Random Generator based on Simon's Two-Stage Design</i>
---------	---

---

**Description**

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

**Usage**

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

**Arguments**

R	positive <a href="#">integer</a> scalar, number of trials $R$
n1, n	positive <a href="#">integer</a> scalars, Stage-1 sample size $n_1$ and total sample size $n$
r1	non-negative <a href="#">integer</a> scalar, number of response in Stage-1 $r_1$ required <i>exclusively</i> . In other words, passing Stage-1 indicates observing $> r_1$ responses
prob	<a href="#">numeric</a> 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))
cut.default(ys, breaks = c(0, 4L, 15L, 54L), right = TRUE,
  labels = c('early-termination', 'fail', 'success'))
```

---

show,Simon\_oc-method    *Show [Simon\\_oc](#) Object*

---

**Description**

Show [Simon\\_oc](#) object

**Usage**

```
## S4 method for signature 'Simon_oc'
show(object)
```

**Arguments**

object            [Simon\\_oc](#) object

**Value**

The [show](#) method for [Simon\\_oc](#) object does not have a returned value.

---

show,Simon\_pr-method    *Show [Simon\\_pr](#) Object*

---

**Description**

Show [Simon\\_pr](#) object

**Usage**

```
## S4 method for signature 'Simon_pr'
show(object)
```

**Arguments**

object            [Simon\\_pr](#) object

**Value**

The [show](#) method for [Simon\\_pr](#) object does not have a returned value.

---

[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	<i>named numeric vector</i> , true response rate(s) of (multiple) drug(s). The names(prob) should be the respective keyword(s) for the drug(s).
simon	<a href="#">ph2simon</a> object
type	<i>character</i> scalar, type of Simon's two-stage design. Currently supports 'minimax' (default) for minimum total sample size, 'optimal' for minimum expected total sample size <i>under</i> $p_0$ , 'n1' for minimum Stage-1 sample size $n_1$ , 'maximax' to use up the user-provided maximum total sample size (parameter nmax of <a href="#">ph2simon</a> )
R	<i>integer</i> scalar, number of simulations. Default 1e4L.
n1, n	(optional) <i>integer</i> scalars, Stage-1 sample size $n_1$ and total sample size $n$ . Will be overridden if simon is given
r1, r	(optional) <i>integer</i> scalars, number of response in Stage-1 $r_1$ and overall $r$ required <i>exclusively</i> . In other words, passing Stage-1 means observing $> r_1$ response. Will be overridden if simon is given
...	potential parameters, currently not in use

**Details**

..

**Value**

[Simon\\_oc](#) returns [Simon\\_oc](#) object

**References**

[doi:10.1016/01972456\(89\)900159](https://doi.org/10.1016/01972456(89)900159)

**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\_oc-class

*S4 Class* [Simon\\_oc](#)


---

**Description**

Operating characteristics of Simon's two-stage design.

**Slots**

.Data [Simon\\_pr](#) object

maxResp [integer vector](#) of length  $p$ , the frequencies of each regime having maximum response. The summation of maxResp is the number of simulation copies.

Simon\_maxResp [integer vector](#) of length  $p$ , the frequencies of each regime having maximum response and success in Simon's two-stage trial.

---

 Simon\_pr

---

*Probabilities associated with 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 [numeric 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*. In other words, 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  $\Pr(X_1 \leq r_1)$ .

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 rejecting  $H_0$  is

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

Parameters nomenclature of n1, n, r1 and r follows that of PASS and [ph2simon](#).

**Value**

`Simon_pr` returns `Simon_pr` object.

**References**

[doi:10.1016/01972456\(89\)900159](https://doi.org/10.1016/01972456(89)900159)

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

**Examples**

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

---

Simon_pr-class	<i>S4 Class <code>Simon_pr</code></i>
----------------	---------------------------------------

---

**Description**

Probabilities of early termination, failure and success, of Simon's Two-Stage Design.

**Slots**

.Data  $l \times 3$  **numeric 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** of length  $l$ , expected sample size(s)  $E(N)$   
 prob **numeric vector** of length  $l$ , response rate(s)  $p$

---

Sprintf.ph2simon	<i>Short Paragraph to Describe a <code>ph2simon</code> Object</i>
------------------	---

---

**Description**

To create a short paragraph to describe a `ph2simon` object.

**Usage**

```
Sprintf.ph2simon(model, type = c("minimax", "optimal", "n1", "maximax"), ...)
```

**Arguments**

model	<b>ph2simon</b> object
type	<b>character</b> scalar, type of Simon's two-stage design, 'minimax' (default) minimum total sample size 'optimal' minimum expected total sample size <i>under</i> $p_0$ 'n1' minimum Stage-1 sample size 'maximax' maximum total sample size (as provided by end-user)
...	additional parameters, currently not in use

**Value**

`Sprintf.ph2simon` returns a [noquote character](#) scalar.

**Examples**

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

---

Sprintf.Simon_oc	<i>Short Paragraph to Describe a <a href="#">Simon_oc</a> Object</i>
------------------	--

---

**Description**

To create a short paragraph to describe a [Simon\\_oc](#) object.

**Usage**

```
Sprintf.Simon_oc(model, ...)
```

**Arguments**

model	<a href="#">Simon_oc</a> object
...	additional parameters, currently not in use

**Value**

`Sprintf.Simon_oc` returns a [noquote character](#) scalar.

---

summary.ph2simon	<i>Summarize a Simon's Two-Stage Design</i>
------------------	---

---

**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**

[summary.ph2simon](#) returns a [list](#) with three (3) elements

'design' [integer matrix](#)

'EN' [double matrix](#)

'p' [double matrix](#)

**Examples**

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

# Index

autoplot.ph2simon, [2](#)  
autoplot.ph2simon(), [2](#)

character, [6](#), [8](#), [9](#)

double, [10](#)

ggplot, [2](#)

integer, [4](#), [6](#), [7](#), [10](#)

list, [10](#)

matrix, [8](#), [10](#)

noquote, [9](#)  
numeric, [4](#), [6–8](#)

ph2simon, [2](#), [3](#), [6–8](#), [10](#)  
print, [3](#)  
print\_ph2simon, [3](#)  
print\_ph2simon(), [3](#)

r\_simon, [3](#), [4](#)  
r\_simon(), [4](#)

show, [5](#)  
show, Simon\_oc-method, [4](#)  
show, Simon\_pr-method, [5](#)  
Simon\_oc, [4](#), [5](#), [5](#), [6](#), [9](#)  
Simon\_oc-class, [6](#)  
Simon\_pr, [5](#), [7](#), [7](#), [8](#)  
Simon\_pr-class, [8](#)  
Sprintf.ph2simon, [8](#), [9](#)  
Sprintf.Simon\_oc, [9](#), [9](#)  
summary.ph2simon, [9](#), [10](#)

vector, [4](#), [6–8](#)