

# Package ‘OptGS’

February 29, 2024

**Type** Package

**Title** Near-Optimal Group-Sequential Designs for Continuous Outcomes

**Version** 1.2

**Date** 2024-02-27

**Maintainer** James Wason <james.wason@newcastle.ac.uk>

**Description** Optimal group-sequential designs minimise some function of the expected and maximum sample size whilst controlling the type I error rate and power at a specified level. ‘OptGS’ provides functions to quickly search for near-optimal group-sequential designs for normally distributed outcomes. The methods used are described in Wason, JMS (2015) <[doi:10.18637/jss.v066.i02](https://doi.org/10.18637/jss.v066.i02)>.

**License** GPL-2

**NeedsCompilation** yes

**Author** James Wason [aut, cre],  
John Burkardt [ctb],  
R O’Neill [ctb]

**Repository** CRAN

**Date/Publication** 2024-02-29 13:02:48 UTC

## R topics documented:

all.class . . . . .	2
optgs . . . . .	2
powerfamily . . . . .	4

<b>Index</b>	<b>6</b>
--------------	----------

---

all.class

*Different generic functions for class OptGS*


---

### Description

Generic functions for summarising an object of class OptGS

### Usage

```
## S3 method for class 'OptGS'
print(x,... )
## S3 method for class 'OptGS'
plot(x,ylim=NULL,...)
```

### Arguments

x	An output object of class OptGS
ylim	y limits to be passed to plot
...	Additional arguments to be passed.

### Details

print.OptGS gives the group-size, stopping boundaries, and operating characteristics of the design  
 plot.OptGS produces a plot of the expected sample size as the standardised treatment effect differs

### Value

Screen or graphics output.

---

optgs

*Finding optimal and balanced group-sequential designs*


---

### Description

optgs is used to find a one-sided multi-stage design that balances four optimality criteria for a RCT with normally distributed outcomes

### Usage

```
optgs(delta0 = 0, delta1 = 1/3, J = 2, sigma = 1, sd.known = TRUE,
      alpha = 0.05, power = 0.9, weights = c(0.95, 0, 0, 0.05),
      initial = NULL)
```

**Arguments**

<code>delta0</code>	mean difference in treatment effect under the null hypothesis (default: 0)
<code>delta1</code>	clinically relevant difference used to power the trial (default: 1/3)
<code>J</code>	number of stages in the trial (default: 2)
<code>sigma</code>	assumed standard deviation of treatment responses (default: 1)
<code>sd.known</code>	logical value indicating if sigma will be treated as known; if FALSE, a quantile substitution method will be used to modify the stopping boundaries (default TRUE)
<code>alpha</code>	one-sided type-I error rate required (default: 0.05)
<code>power</code>	power required (default: 0.9)
<code>weights</code>	vector of length 4 giving the weights put on the four optimality criteria (default: <code>c(0.95,0,0,0.05)</code> ). See details for more information
<code>initial</code>	starting values for the Nelder-Mead algorithm if the user wishes to override the default (default: NULL). Initial values must be specified as a two-dimensional vector where both entries are between -0.5 and 0.5.

**Details**

optgs uses the extended power-family of group-sequential tests, and searches for the values of the futility and efficacy shape parameters that optimise the specified weighting. A description of the extended power-family and optgs is provided in Wason (2012). The ‘weights’ argument corresponds to the weight put on: 1) the expected sample size at  $\delta=\delta_0$ ; 2) the expected sample size at  $\delta=\delta_1$ ; 3) the maximum expected sample size; 4) the maximum sample size (i.e.  $J*\text{groupsize}$ ).

**Value**

<code>groupsize</code>	the number of patients required per arm, per stage
<code>futility</code>	the futility boundaries for the design
<code>efficacy</code>	the efficacy boundaries for the design
<code>ess</code>	the expected sample size at the $\delta_0$ ; the expected sample size at the $\delta_1$ ; and the maximum expected sample size
<code>typeIerror</code>	the actual type-I error rate of the design
<code>power</code>	the actual power of the design

**References**

Wason, J.M.S. OptGS: an R package for finding near-optimal group-sequential designs. *Journal of Statistical Software*, 66(2), 1-13. <https://www.jstatsoft.org/v66/i02/>

**Examples**

```
##Find a three-stage design that minimises the maximum expected sample size.
threestagedeltaminimax=optgs(J=3,weights=c(0,0,1,0))
plot(threestagedeltaminimax)
```

powerfamily

*Finding extended power-family group-sequential designs***Description**

powerfamily is used to find a one-sided extended power-family group-sequential design

**Usage**

```
powerfamily(futility = 0, efficacy = 0, delta0 = 0, delta1 = 1/3,
            J = 2, sigma = 1, sd.known = TRUE, alpha = 0.05, power = 0.9)
```

**Arguments**

futility	shape parameter for futility boundaries (default: 0)
efficacy	shape parameter for efficacy boundaries (default: 0)
delta0	mean difference in treatment effect under the null hypothesis (default: 0)
delta1	clinically relevant difference used to power the trial (default: 1/3)
J	number of stages in the trial (default: 2)
sigma	assumed standard deviation of treatment responses (default: 1)
sd.known	logical value indicating if sigma will be treated as known; if FALSE, a quantile substitution method will be used to modify the stopping boundaries (default TRUE)
alpha	one-sided type-I error rate required (default: 0.05)
power	power required (default: 0.9)

**Details**

powerfamily uses the extended power-family of group-sequential tests. A description of the extended power-family is provided in Wason (2012).

**Value**

groupsize	the number of patients required per arm, per stage
futility	the futility boundaries for the design
efficacy	the efficacy boundaries for the design
ess	the expected sample size at the delta0; the expected sample size at the delta1; and the maximum expected sample size
typeIerror	the actual type-I error rate of the design
power	the actual power of the design

**References**

Wason, J.M.S. OptGS: an R package for finding near-optimal group-sequential designs. *Journal of Statistical Software*, 66(2), 1-13. <http://www.jstatsoft.org/v66/i02/>

**Examples**

```
##Find a three-stage design that has shape parameters -0.5 and 0.5.  
threestagedesign=powerfamily(J=3,futility=-0.5,efficacy=0.5)  
plot(threestagedesign)
```

# Index

`all.class`, [2](#)

`optgs`, [2](#)

`plot.OptGS (all.class)`, [2](#)

`powerfamily`, [4](#)

`print.OptGS (all.class)`, [2](#)