Package 'tsdf'

July 22, 2025

Type Package

Title Two-/Three-Stage Designs for Phase 1&2 Clinical Trials

Version 1.1-8

Date 2020-05-09

Author Wenchuan Guo, Jianan Hui, Bob Zhong

Maintainer Wenchuan Guo <wguo1017@gmail.com>

Description Calculate optimal Zhong's two-/three-stage Phase II designs (see Zhong (2012) <doi:10.1016/j.cct.2012.07.006>). Generate Target Toxicity decision table for Phase I dose-finding (two-/three-stage). This package also allows users to run dose-finding simulations based on customized decision table.

License GPL-2

Encoding UTF-8

LazyData true

RoxygenNote 7.0.2

Suggests knitr

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2020-05-09 15:40:02 UTC

Contents

`adj.two` *Zhong's 2-/3- stage Phase II design*

Description

adjust Zhong's 2-/3-stage design for over-/under-running

Usage

```
adj.two(n1, r1, s1, n2, alpha1, alpha2, beta, pc, pe, ...)
```

Arguments

`n1`	sample size at stage 1.
`r1`	inefficacy boundary at stage 1.
`s1`	efficacy boundary at stage 1. if no early stopping for efficacy, `s1` should equal to `n1`.
`n2`	sample size at stage 2.
`alpha1`	left-side overall type I error.
`alpha2`	right-side overall type I error.
`beta`	type II error.
`pc`	a numeric vector of response rate. should be a vector with length 1 or 2.
`pe`	alternative hypothesis.
`...`	not used argument.

Details

To be added

Value

An object of class "opt.design" is a list containing:

`bdry`	rejection regions
`error`	true type 1/2 errors
`n`	sample size at each stage
`complete`	complete list of feasible designs
`alpha1`	input; left-side type 1 error
`alpha2`	input; right-side type 1 error
`beta`	input; type 2 error
`pc`	input; a vector of response rate.
`pe`	input; a vector of alternative response rate
`sf`	input; the alpha-spending function used
`stage`	input; two- or three- stage design is used

Author(s)

Wenchuan Guo <wguo1017@gmail.com>, Jianan Hui <jiananhuistat@gmail.com>

Examples

```
n1 <- 22
r1 <- 6
s1 <- 22
n2 <- 24
pc <- 0.4
pe <- pc + 0.15
alpha1 <- 0.3
alpha2 <- 0.1
beta <- 0.2
out <- adj.two(n1, r1, s1, n2, alpha1, alpha2, beta, pc, pe)
```

`dec.sim` *run dose-finding simulations*

Description

Run dose-finding simulations based on a customized decision table.

Usage

```
dec.sim(truep, decTable, start.level = 1, nsim = 1000)
```

Arguments

`truep`	a vector of length k (the number of doses being considered in the trial), with values equal to the true probabilities of toxicity at the dose levels.
`decTable`	a customized decision table. (same format as output of `dec.table`)
`start.level`	starting dose level. Defaults to 1, i.e. the lowest dose level.
`nsim`	the number of simulation trials. Defaults to 1000.

Details

Assume there are d dose levels to be studied. Denote the cumulative number of patients treated and cumulative number of DLTs at the current dose level are n_i and m_i, respectively. n_max is the maximum number of patients permitted to be treated at each dose level. The procedure is as follows

- Step 1 : Update cumulative number of DLTs m_i and total number of patients n_i treated at the current dose and use the decision table to make a decision: if decision is “S" –> step 2; if decision is “D" or “DU” –> step 3; if decision is “E” –> step 4
- Step 2 : If $n_i = n_max$, declare dose i as the MTD; otherwise, update m_i and n_i with additional cohort of patients and go to Step 1.

- Step 3 : If the current dose level is the highest dose level, then: if n_i < n_max, update m_i and n_i with additional cohort of patients and go to Step 1; otherwise, stop the trial and declare that the MTD is higher than the highest dose level (inconclusive); If the current dose is not the lowest dose, then: if $n_i-1 < n_max$, update m_i-1 and n_i-1 with additional cohort of patients and set the current dose level to be the next lower dose level, and go to Step 1; otherwise, stop the trial and declare the next lower dose level is the MTD; Additionally, if the decision is "DU", record this dose level as DU and never treat additional patients at the current dose level again.
- Step 4 : If the current dose level is the highest dose level, then: if $n_i < n_max$, update m_i and n_i with additional cohort of patients and go to Step 1; otherwise, stop the trial and declare that the MTD is higher than the highest dose level (inconclusive); If the next higher dose level is of status DU, then: if $n_i < n_max$, update m_i and n_i with additional cohort of patients and go to step 1; otherwise stop, the current dose level is MTD; Otherwise: if $n_i+1 < n_max$, update m_i+1 and n_i+1 with additional cohort of patients, set the current dose level to be next higher dose level, and go to step 1; else, the current dose level is the MTD.

Value

the functions `summary.dec.sim` is used to obtain and print a summary table of the results (recommended). An object of class `"dec.sim"` is a list containing:

`mtd`	a vector of dose levels giving the recommended maximum tolerated dose (MTD) at the end of the trial.
`mtd.prob`	a vector of length `k` giving the average proportions of selected as MTD at each dose level.
`over.prob`	a vector of length `k` giving the average proportions of selected as over the MTD at each dose level.
`n.patients`	the average number of patients dosed at each level.
`dlt`	the average number of DLTs experienced at each dose level.
`truep`	input; true probabilities of toxicity.
`start.level`	input; starting dose level.
`nsim`	input; number of simulated trails.

Author(s)

Wenchuan Guo <wguo1017@gmail.com>

Examples

```
truep <- c(0.3, 0.45, 0.5, 0.6)
res <- dec.table(0.6,0.4,0.2,0.3,c(3,3,3))
out <- dec.sim(truep, res$table, start.level = 2, nsim = 1000)
summary(out, pt = 0.3)
```

`dec.table` *generate three-stage dose-finding decision table*

Description

Generate three stage dose finding decision table

Usage

```
dec.table(alpha.l, alpha.r, alpha.u, pt, n, sf.param = 4, pe.par = 0.25, ...)
```

Arguments

`alpha.l`	left-side overall type 1 error. Control the upper bound of dose escalation.
`alpha.r`	right-side overall type 1 error. Control the lower bound of dose de-escalatition.
`alpha.u`	right-side overall type 1 error. This also controls the lower bound of dose de-escalatition, but it is used to find lower bound for "DU".
`pt`	a numeric vector of target toxicity. Should be a vector with 1 or 2(when the target is an interval).
`n`	a vector of sample size at each stage. `sum(n)` is the total sample size. For A+B designs, `n` is a vector with length 2; for A+B+C designs, `n` has length 3.
`sf.param`	a single real value specifying the gamma parameter for which Hwang-Shih-DeCani spending is to be computed; allowable range is [-40, 40]. Increasing this parameter implies that more error is spent early stage and less is available in late stage. Default to 4.
`pe.par`	alternative hypothesis that used to calculate power/type 2 error. The alternative is set to be `pe = pt + pe.par`. Default to 0.25.
`...`	not used argument.

Details

Alpha-spending method is added to two-/three-stage designs. `dec.table` supports Hwang-Shih-DeCani spending function.

Value

An object of class "dec.table" is a list containing:

`table`	the generated decision table.
`alpha.two`	a vector of true type 1 error for two-tailed test.
`alpha.one`	a vector of true type 1 error for right-tailed test.
`beta`	a single value of true type 2 error(depends on alternative).
`E`	a vector of "E" bound.
`D`	a vector of "D" bound.

DU	a vector of "DU" bound.
pt	input; a vector of target toxicity
n	input; a vector with sample size at each stage.
sf.param	input; the alpha-spending function parameter used.

Author(s)

Wenchuan Guo <wguo007@ucr.edu>

Examples

```
alpha.l <- 0.6
alpha.r <- 0.4
alpha.u <- 0.1
pt <- 0.3
# print out decision table for a 3+3+3 design
n <- rep(3, 3)
dec.table(alpha.l, alpha.r, alpha.u, pt, n)$table
# 3+3 design
n <- rep(3, 2)
dec.table(alpha.l, alpha.r, alpha.u, pt, n)$table
```

---

opt.design *Zhong's 2-/3- stage Phase II design*

Description

calculate optimal 2-/3-stage design given by Bob Zhong

Usage

```
opt.design(
  alpha1,
  alpha2,
  beta,
  pc,
  pe,
  stage = 2,
  stop.eff = FALSE,
  frac_n1 = NULL,
  frac_n2 = NULL,
  sf.param = NULL,
  show = FALSE,
  nmax = 100,
  n.choice = 1,
  ...
)
```

Arguments

alpha1	left-side overall type I error.
alpha2	right-side overall type I error.
beta	type II error
pc	a numeric vector of response rate. should be a vector with length 1 or 2.
pe	alternative hypothesis.
stage	2 or 3. default to 2 (2-stage design).
stop.eff	logical flag. default to FALSE. if stop.eff = TRUE, the trial may stop for efficacy at interim.
frac_n1	proportion of n1. for 2-stage design, default to c(0.3, 0.6), i.e. the range of n1 is 0.2*n to 0.5*n. for 3-stage design, default to c(0.2, 0.3), i.e. the range of n1 is 0.2*n to 0.3*n
frac_n2	proportion of n2. Used for 3-stage design. default to c(0.2, 0.4).
sf.param	a single real value specifying the gamma parameter for which Hwang-Shih-DeCani spending is to be computed; allowable range is [-40, 40]. Increasing this parameter implies that more error is spent early stage and less is available in late stage. For two-stage designs, default to NULL(alpha-spending is not used); for three-stage designs, default to 4.
show	logical. If TRUE, current sample size is shown as total sample size increase.
nmax	maximum sample size. default to 100.
n.choice	stop criterion for the search of feasible designs. stop if number of designs is more than n.choice
...	not used argument.

Details

The two-stage design setup is: n1 patients are treated in the first stage. At the end of the first stage, either the trial continues to the second stage or inefficacy is concluded and the trial is stopped (early termination), depending on the number of responses observed at the first stage. If the trial does continue to the second stage, additional n2 patients are treated. Three-stage design is an extension of two-stage design where one stage is added between Stage 1 and 2. The left-side rejection region is response <= r_i for i = 1, 2, or 3 and right-side rejection region is response > s. Alpha-spending method is added to two-/three-stage designs. opt.design supports Hwang-Shih-DeCani spending function. You can change the definition of HSD function to use a different spending function.

Value

An object of class "opt.design" is a list containing:

bdry	rejection regions
error	true type 1/2 errors
n	sample size at each stage
complete	complete list of feasible designs

alpha1	input; left-side type 1 error
alpha2	input; right-side type 1 error
beta	input; type 2 error
pc	input; a vector of response rate.
pe	input; a vector of alternative response rate
sf	input; the alpha-spending function used
stage	input; two- or three- stage design is used

Author(s)

Wenchuan Guo <wguo1017@gmail.com>, Jianan Hui <jiananhuistat@gmail.com>

References

Zhong. (2012) Single-arm Phase IIA clinical trials with go/no-go decisions. *Contemporary Clinical Trials*, **33**, 1272–1279.

Examples

```
alpha1 <- 0.15
alpha2 <- 0.10
beta <- 0.15
pc <- 0.25
pe <- pc + 0.20
# calculate optimal two-stage design without using alpha-spending
opt.design(alpha1, alpha2, beta, pc, pe, stage=2)
## Not run:
# calculate optimal two-stage design with Pocock-like spending function
opt.design(alpha1, alpha2, beta, pc, pt, stage = 2, sf.param = 1)

# calculate optimal three-stage design with =Oâ€™Brien-Fleming like spending function
opt.design(alpha1, alpha2, beta, pc, pt, stage = 3, sf.param = -4)

## End(Not run)
```

---

plot.dec.sim *plot simulation results from a dec.sim object*

Description

Three plots are currently available: a plot of true toxicity at each dose level (type = "s"); a bar plot of the probability of selecting as the MTD for each dose level (type = "prob"); a bar plot of the average number of patients treated at each dose level (type = "np"); a bar plot of the average number of patients experienced DLT at each dose level (type = "dlt") and type = "all" generates all above plots.

Usage

```
## S3 method for class 'dec.sim'
plot(
  x,
  pt,
  s = 1,
  type = c("all", "s", "prob", "np", "dlt"),
  label = TRUE,
  col = "cornflowerblue",
  text.col = "darkblue",
  cex = 1,
  ...
)
```

Arguments

x	an object of class "dec.sim" or "sl.sim", a result of a call to dec.sim or sl.sim.
pt	a vector with target toxicity for each scenario.
s	scenario to be plotted. Defaults to 1.
type	plot type. See descriptions above.
label	a logical value indicating if values are shown on plot.
col	graphical parameter col; see details par.
text.col	plotting color of text shown.
cex	graphical parameter col; see details par.
...	arguments to be passed to plot methods.

Examples

```
# generate decision table
dt <- dec.table(0.6,0.4,0.2,0.3,c(3,3,3))
# simulate trials from test data
test.file <- system.file("extdata", "testS.csv", package = "tsdf")
out <- sl.sim(dt$table, test.file)
plot(out, pt=rep(0.3,2), s=1, type="all")
plot(out, pt=rep(0.3,2), s=2, type="prob")
plot(out, pt=rep(0.3,2), s=1, type="np")
plot(out, pt=rep(0.3,2), s=2, type="dlt")
```

---

plot.dec.table *plot decision table from a "dec.table" object.*

Description

plot method for class "dec.table"

Usage

```
## S3 method for class 'dec.table'
plot(x, ...)
```

Arguments

x	an object of class "dec.table", a result of a call to dec.table.
...	Not used argument.

Details

plot.dec.table prints the decision boundarys.

Examples

```
truep <- c(0.3, 0.45, 0.5, 0.6)
out <- dec.table(0.6,0.4,0.2,0.3,c(3,3,3))
plot(out)
```

---

print.dec.table *print decision table from a "dec.table" object.*

Description

print method for class "dec.table"

Usage

```
## S3 method for class 'dec.table'
print(x, ...)
```

Arguments

x	an object of class "dec.table", a result of a call to dec.table.
...	Not used argument.

Details

print.dec.table prints the decision table with legend keys.

Examples

```
print(dec.table(0.6,0.4,0.2,0.3,c(3,3,3)))
```

print.opt.design *print Zhong's design from a "opt.design" object.*

Description

print method for class "opt.design"

Usage

```
## S3 method for class 'opt.design'
print(x, ...)
```

Arguments

x	an object of class "opt.design", a result of a call to opt.design.
...	not used argument.

Examples

```
alpha1 <- 0.20
alpha2 <- 0.1
beta <- 0.20
pc <- 0.5
pt <- pc + 0.2
out <- opt.design(alpha1, alpha2, beta, pc, pt, stage = 2, sf.param = 1)
print(out)
```

sl.sim *Dose-finding simulations for a list of scenarios*

Description

Run dose-finding simulations based on a customized decision table for a list of scenarios.

Usage

```
sl.sim(decTable, file, header = TRUE, sep = ",", ...)
```

Arguments

decTable	A customized decision table. (same format as output of dec.table)
file	The name of the file which the data are to be read from. See details in read.table.
header	A logical value indicating whether the file contains the names of the variables as its first line. Default is FALSE. See details in read.table.
sep	The field separator character. Default is ",". See details in read.table.
...	arguments to be passed to read.table methods.

Details

In each line of the input file, the parameters must be ordered in accordance as follows: `pt`, `start.level`, `nsim`, `truep`. See details in `read.table`. The algorithm for dose-finding is described in `dec.sim`.

Value

The function `summary` is used to obtain and print a summary table of the results. An object of class `"dec.sim"` (1 scenario) or `"sl.sim"` (more than 1 scenarios)is a list containing:

`MTD`	A vector of dose levels giving the recommended maximum tolerated dose (MTD) at the end of the trial.
`n.patients`	The average number of patients dosed at each level.
`truep`	input; true probabilities of toxicity.
`start.level`	input; starting dose level.
`nsim`	input; number of simulated trails.

Author(s)

Wenchuan Guo <wguo007@ucr.edu>

Examples

```
dt <- dec.table(0.6,0.4,0.2,0.3,c(3,3,3))
test.file <- system.file("extdata", "testS.csv", package = "tsdf")
# use a customized decision table
table.file <- system.file("extdata", "decTable.csv", package = "tsdf")
dec <- read.table(table.file, sep=",", col.names=c(3,4,8,10), row.names = 1, check.names = FALSE)
out1 <- sl.sim(dt$table, test.file)
out2 <- sl.sim(dec, test.file)
```

---

summary.dec.sim *Summarizing simulation results from a dec.sim object*

Description

`summary` method for class `"dec.sim"`.

Usage

```
## S3 method for class 'dec.sim'
summary(object, pt, ...)
```

Arguments

`object`	an object of class `"dec.sim"`, a result of a call to `dec.sim` or `sl.sim`.
`pt`	target toxicity for each scenario.
`...`	Not used argument.

Details

summary is used for formating important statistics for dose-finding simulation. Giving the target toxicity, it returns the probability of selecting current dose level as the MTD and over the MTD, probability of selecting the true MTD, probability of subjects treated at or below the true MTD, etc. The MTD is defined as the highest dose level such that the toxicity probability is less than target toxicity probability, if target is less than the smallest probability, then the lowest dose level is set as MTD. For example, if target is 0.3 and true toxicity for five doses are 0.1, 0.25, 0.35, 0.40, then MTD is dose 2.

Examples

```
test.file <- system.file("extdata", "testS.csv", package = "tsdf")
dt <- dec.table(0.6,0.4,0.2,0.3,c(3,3,3))
out <- sl.sim(dt$table, test.file)
pt <- c(0.3, 0.4)
summary(out, pt)
```

Index