Package ‘ASSISTant’

May 3, 2019

Type Package

Title Adaptive Subgroup Selection in Group Sequential Trials

Version 1.4.2

Date 2019-05-01

VignetteBuilder knitr

URL https://github.com/bnaras/ASSISTant

BugReports https://github.com/bnaras/ASSISTant/issues

Description Clinical trial design for subgroup selection in three-stage group sequential trial. Includes facilities for design, exploration and analysis of such trials. An implementation of the initial DEFUSE-3 trial is also provided as a vignette.

License GPL (>= 2)

Encoding UTF-8

RoxygenNote 6.1.1

Imports R6, mvtnorm, knitr, magrittr, dplyr

Suggests rmarkdown

NeedsCompilation no

Author Tze Leung Lai [ctb],
Philip Lavori [aut],
Olivia Liao [aut],
Balasubramanian Narasimhan [aut, cre],
Ka Wai Tsang [aut]

Maintainer Balasubramanian Narasimhan <naras@stat.Stanford.EDU>

Repository CRAN

Date/Publication 2019-05-03 19:10:03 UTC

R topics documented:

`ASSISTant` *Three stage group sequential adaptive design with subgroup selection*

Description

`ASSISTant` is a package that implements a three-stage adaptive clinical trial design with provision for subgroup selection where the treatment may be effective. The main design object is an `R6` class that can be instantiated and manipulated to obtain the operating characteristics. A vignette is provided showing the use of this package for designing the DEFUSE-3 trial, described in the paper by Lai, Lavori and Liao. The package contains everything necessary to reproduce the results of the paper.

References

Adaptive Choice of Patient Subgroup for Comparing Two Treatments by Tze Leung Lai and Philip W. Lavori and Olivia Yueh-Wen Liao. Contemporary Clinical Trials, Vol. 39, No. 2, pp 191-200 (2014). `http://www.sciencedirect.com/science/article/pii/S1551714414001311`

Adaptive design of confirmatory trials: Advances and challenges, `http://www.sciencedirect.com/science/article/pii/S1551714415300239` by Tze Leung Lai and Philip W. Lavori and Ka Wai Tsang. Contemporary Clinical Trials, Vol. 45, Part A, pp 93-102 (2015).

`ASSISTDesign` *A class to encapsulate the adaptive clinical trial design of Lai, Lavori and Liao*

Description

`ASSISTDesign` objects are used to design, simulate and analyze adaptive group sequential clinical trial with three stages.

Usage

```
# design <- ASSISTDesign$new(trialParameters, designParameters)
```

Format

An `R6Class` generator object

Methods

`ASSISTDesign$new(designParameters, trialParameters, discreteData = FALSE, boundaries)`
: Create a new `ASSISTDesign` instance object using the parameters specified. If `discreteData` is `TRUE` use a discrete distribution for the Rankin scores and `designParameters` must contain the appropriate distributions to sample from. If `boundaries` is specified, it used.

`getDesignParameters`,`getTrialParameters`, `getBoundaries` Accessor methods for (obvious) object fields

`setBoundaries` Modifier method for boundaries a named vector of double values with names `btilde`, `b`, and `c`, in that order

`print()` Print the object in a human readable form

`computeCriticalValues()` Compute the critical boundary values $\tilde{b}$, b and c for futility, efficacy and final efficacy decisions; saved in field `boundaries`

`explore(numberOfSimulations = 5000, rngSeed = 12345)` Explore the design using the specified number of simulations and random number seed. There are a number of further parameters. By default `trueParameters = self$getDesignParameters()` as would be the case for a Type I error calculation. If changed, would yield power. Also `recordStats = TRUE/FALSE`, `showProgress = TRUE/FALSE`, `saveRawData = TRUE/FALSE` control recording statistics, raw data saves, display of progress. Fixed sample size (`fixedSampleSize = TRUE/FALSE`) can be specified to ensure that patients lost after a futile overall look are not made up. Returns a list of results

`analyze(trialExploration)` Analyze the design given the `trialExploration` which is the result of a call to `explore` to simulate the design. Return a list of summary quantities

`summary(analysis)` Print the operating characteristics of the design, using the analysis result from the `analyze` call

References

Adaptive Choice of Patient Subgroup for Comparing Two Treatments by Tze Leung Lai and Philip W. Lavori and Olivia Yueh-Wen Liao. Contemporary Clinical Trials, Vol. 39, No. 2, pp 191-200 (2014). doi:10.1016/j.cct.2014.09.001g

See Also

LLL.SETTINGS for an explanation of trial parameters

Examples

```
## Not run:
data(LLL.SETTINGS)
prevalence <- LLL.SETTINGS$prevalences$table1
scenario <- LLL.SETTINGS$scenarios$S0
designParameters <- list(prevalence = prevalence,
                       mean = scenario$mean,
                       sd = scenario$sd)
designA <- ASSISTDesign$new(trialParameters = LLL.SETTINGS$trialParameters,
                            designParameters = designParameters)
print(designA)
## A realistic design uses 5000 simulations or more!
result <- designA$explore(showProgress = interactive())
analysis <- designA$analyze(result)
designA$summary(analysis)

## End(Not run)
## For full examples, try:
## browseURL(system.file("full_doc/ASSISTant.html", package="ASSISTant"))
```

ASSISTDesignB *A fixed sample design to compare against the adaptive clinical trial design of Lai, Lavori and Liao.*

Description

ASSISTDesignB objects are used to design a trial with certain characteristics provided in the object instantiation method. This design differs from ASSISTDesign in only how it computes the critical boundaries, how it performs the interim look, and what quantities are computed in a trial run.

Usage

```
# design <- ASSISTDesignB$new(trialParameters, designParameters, discreteData)
```

Format

An R6Class generator object

Methods

`ASSISTDesignB$new(designParameters, trialParameters, discreteData = FALSE, boundaries)` Create a new `ASSISTDesign` instance object using the parameters specified. If `discreteData` is `TRUE` use a discrete distribution for the Rankin scores and `designParameters` must contain the appropriate distributions to sample from. If `boundaries` is specified, it is used

`getDesignParameters`,`getTrialParameters`, `getBoundaries` Accessor methods for (obvious) object slots

`setBoundaries` Modifier method for boundaries a named vector of double values with names `btilde`, `b`, and `c`, in that order

`print()` Print the object in a human readable form

`computeCriticalValues()` Compute the critical boundary value c_α

`explore(numberOfSimulations = 5000, rngSeed = 12345)` Explore the design using the specified number of simulations and random number seed. There are further parameters. By default `trueParameters = self$getDesignParameters()` as would be the case for a Type I error calculation. If changed, would yield power. Also `showProgress = TRUE/FALSE`, `saveRawData = TRUE/FALSE` control raw data saves and display of progress. Returns a list of results

`analyze(trialExploration)` Analyze the design given the `trialExploration` which is the result of a call to `explore` to simulate the design. Return a list of summary quantities

`summary(analysis)` Print the operating characteristics of the design, using the analysis result from the `analyze` call

References

Adaptive Choice of Patient Subgroup for Comparing Two Treatments by Tze Leung Lai and Philip W. Lavori and Olivia Yueh-Wen Liao. Contemporary Clinical Trials, Vol. 39, No. 2, pp 191-200 (2014). doi:10.1016/j.cct.2014.09.001g

See Also

`ASSISTDesign` which is a superclass of this object

Examples

```
## Not run:
data(LLL.SETTINGS)
prevalence <- LLL.SETTINGS$prevalences$table1
scenario <- LLL.SETTINGS$scenarios$S0
designParameters <- list(prevalence = prevalence,
                       mean = scenario$mean,
                       sd = scenario$sd)
designB <- ASSISTDesignB$new(trialParameters = LLL.SETTINGS$trialParameters,
                            designParameters = designParameters)
print(designB)
## A realistic design uses 5000 simulations or more!
result <- designB$explore(showProgress = interactive())
analysis <- designB$analyze(result)
designB$summary(analysis)
```

```
## End(Not run)
## For full examples, try:
## browseURL(system.file("full_doc/ASSISTant.html", package="ASSISTant"))
```

ASSISTDesignC *A fixed sample RCT design to compare against the adaptive clinical trial design of Lai, Lavori and Liao.*

Description

ASSISTDesignC objects are used to design a trial with certain characteristics provided in the object instantiation method. This design differs from ASSISTDesign in only how it computes the critical boundaries, how it performs the interim look, and what quantities are computed in a trial run.

Usage

```
# design <- ASSISTDesignC$new(trialParameters, designParameters)
```

Format

An R6Class generator object

Methods

ASSISTDesignC$new(designParameters, trialParameters, discreteData = FALSE, boundaries)
: Create a new ASSISTDesign instance object using the parameters specified. If discreteData is TRUE use a discrete distribution for the Rankin scores and designParameters must contain the appropriate distributions to sample from. If ‘boundaries is specified, it is used.

getDesignameters,getTrialParameters, getBoundaries Accessor methods for (obvious) object slots

setBoundaries Modifier method for boundaries a named vector of double values with names btilde, b, and c, in that order

print() Print the object in a human readable form

computeCriticalValues() Compute the critical boundary value c_α

explore(numberOfSimulations = 5000, rngSeed = 12345 Explore the design using the specified number of simulations and random number seed. There are further parameters. By default trueParameters = self$getDesignParameters() as would be the case for a Type I error calculation. If changed, would yield power. Also showProgress = TRUE/FALSE, saveRawData = TRUE/FALSE control raw data saves and display of progress. Returns a list of results

analyze(trialExploration) Analyze the design given the trialExploration which is the result of a call to explore to simulate the design. Return a list of summary quantities

summary(analysis) Print the operating characteristics of the design, using the analysis result from the analyze call

References

Adaptive Choice of Patient Subgroup for Comparing Two Treatments by Tze Leung Lai and Philip W. Lavori and Olivia Yueh-Wen Liao. Contemporary Clinical Trials, Vol. 39, No. 2, pp 191-200 (2014). doi:10.1016/j.cct.2014.09.001g

See Also

`ASSISTDesignB` which is a superclass of this object

Examples

```
data(LLL.SETTINGS)
prevalence <- LLL.SETTINGS$prevalences$table1
scenario <- LLL.SETTINGS$scenarios$S0
designParameters <- list(prevalence = prevalence,
                       mean = scenario$mean,
                       sd = scenario$sd)
## A realistic design uses 5000 simulations or more!
designC <- ASSISTDesignC$new(trialParameters = LLL.SETTINGS$trialParameters,
                            designParameters = designParameters)
print(designC)
result <- designC$explore(numberOfSimulations = 100, showProgress = interactive())
analysis <- designC$analyze(result)
designC$summary(analysis)
## For full examples, try:
## browseURL(system.file("full_doc/ASSISTant.html", package="ASSISTant"))
```

`colNamesForStage` *Return a vector of column names for statistics for a given stage*

Description

Return a vector of column names for statistics for a given stage

Usage

```
colNamesForStage(stage, J)
```

Arguments

`stage`	the trial stage (1 to 3 inclusive).
`J`	the number of subgroups

Value

a character vector of the column names

`computeMeanAndSD` *Compute the mean and sd of a discrete Rankin distribution*

Description

Compute the mean and sd of a discrete Rankin distribution

Usage

```
computeMeanAndSD(probVec = rep(1, 7L), support = 0L:6L)
```

Arguments

`probVec`	a probability vector of length equal to length of support, default is uniform
`support`	a vector of support values (default 0:6 for Rankin Scores)

Value

a named vector of `mean` and `sd`

`computeMHPBoundaries` *Compute the three modified Haybittle-Peto boundaries*

Description

Compute the three modified Haybittle-Peto boundaries

Usage

```
computeMHPBoundaries(prevalence, N, alpha, beta, eps,
  futilityOnly = FALSE)
```

Arguments

`prevalence`	the vector of prevalences between 0 and 1 summing to 1. J, the number of groups, is implicitly the length of this vector and should be at least 2.
`N`	a three-vector of total sample size at each stage
`alpha`	the type I error
`beta`	the type II error
`eps`	the fraction (between 0 and 1) of the type 1 error to spend in the interim stages 1 and 2
`futilityOnly`	a logical value indicating only the futility boundary is to be computed; default `FALSE`

Value

a named vector of three values containing $\tilde{b}$, b, c

References

Adaptive Choice of Patient Subgroup for Comparing Two Treatments by Tze Leung Lai and Philip W. Lavori and Olivia Yueh-Wen Liao. Contemporary Clinical Trials, Vol. 39, No. 2, pp 191-200 (2014). http://www.sciencedirect.com/science/article/pii/S1551714414001311

`computeMHPBoundaryITT` *Compute the three modified Haybittle-Peto boundaries and effect size*

Description

Compute the three modified Haybittle-Peto boundaries and effect size

Usage

```
computeMHPBoundaryITT(prevalence, alpha)
```

Arguments

`prevalence`	the vector of prevalences between 0 and 1 summing to 1. J, the number of groups, is implicitly the length of this vector and should be at least 2.
`alpha`	the type I error

Value

a named vector of a single value containing the value for `c`

References

Adaptive Choice of Patient Subgroup for Comparing Two Treatments by Tze Leung Lai and Philip W. Lavori and Olivia Yueh-Wen Liao. Contemporary Clinical Trials, Vol. 39, No. 2, pp 191-200 (2014). http://www.sciencedirect.com/science/article/pii/S1551714414001311

DEFUSE3Design *The DEFUSE3 design*

Description

DEFUSE3Design is a slight variant of the the adaptive clinical trial design of Lai, Lavori and Liao. Simulation is used to compute the expected maximum sample size and the boundary for early futility is adjusted to account as well.

Usage

```
# design <- DEFUSE3Design$new(designParameters, trialParameters)
```

Format

An R6Class generator object

Methods

`DEFUSE3Design$new(designParameters, trialParameters, discreteData = FALSE, numberOfSimulations = 50`
: Create a new DEFUSE3Design instance object using the parameters specified. If discreteData is TRUE use a discrete distribution for the Rankin scores and designParameters must contain the appropriate distributions to sample from. If boundaries is specified, it is used.

`getDesignParameters`,`getTrialParameters`, `getBoundaries` Accessor methods for (obvious) object slots

`setBoundaries` Modifier method for boundaries a named vector of double values with names btilde, b, and c, in that order

`print()` Print the object in a human readable form

`adjustCriticalValues(numberOfSimulations, rngSeed, showProgress)` Adjust the critical values by performing simulations using the parameters provided

`computeCriticalValues()` Compute the critical boundary value c_α

`explore(numberOfSimulations = 5000, rngSeed = 12345, trueParameters = self$getDesignParameters(), r`
: Explore the design using the specified number of simulations and random number seed. trueParameters is by default the same as designParameters as would be the case for a Type I error calculation. If changed, would yield power. Record statistics, save raw data and show progress if so desired. Returns a list of results

`analyze(trialHistory)` Analyze the design given the trialHistory which is the result of a call to explore to simulate the design. Return a list of summary quantities

`summary(analysis)` Print the operating characteristics of the design, using the analysis result from the analyze call

References

Adaptive design of confirmatory trials: Advances and challenges, 2015 45(Pt A):93-102, by Tze Leung Lai and Philip W. Lavori and Ka Wai Tsang. doi:10.1016/j.cct.2015.06.007

See Also

ASSISTDesign which is a superclass of this object

Examples

```
trialParameters <- list(N = c(200, 340, 476), type1Error = 0.025,
                        eps = 1/2, type2Error = 0.1)
designParameters <- list(
   nul0 = list(prevalence = rep(1/6, 6), mean = matrix(0, 2, 6),
               sd = matrix(1, 2, 6)),
   alt1 = list(prevalence = rep(1/6, 6), mean = rbind(rep(0, 6),
               c(0.5, 0.4, 0.3, 0, 0, 0)),
               sd = matrix(1, 2, 6)),
   alt2 = list(prevalence = rep(1/6, 6), mean = rbind(rep(0, 6),
               c(0.5, 0.5, 0, 0, 0, 0)),
               sd = matrix(1,2, 6)),
   alt3 = list(prevalence = rep(1/6, 6), mean = rbind(rep(0, 6), rep(0.36, 6)),
               sd = matrix(1,2, 6)),
   alt4 = list(prevalence = rep(1/6, 6), mean = rbind(rep(0, 6), rep(0.30, 6)),
               sd = matrix(1,2, 6)),
   alt5 = list(prevalence = rep(1/6, 6), mean = rbind(rep(0, 6),
               c(0.4, 0.3, 0.2, 0, 0, 0)),
               sd = matrix(1,2, 6)),
   alt6 = list(prevalence = rep(1/6, 6), mean = rbind(rep(0, 6),
               c(0.5, 0.5, 0.3, 0.3, 0.1, 0.1)),
               sd = matrix(1,2, 6)))

## Not run:
## A realistic design uses 5000 simulations or more!
defuse3 <- DEFUSE3Design$new(trialParameters = trialParameters,
                             numberOfSimulations = 25,
                             designParameters = designParameters$nul0,
                             showProgress = FALSE)
print(defuse3)
result <- defuse3$explore(showProgress = interactive())
analysis <- defuse3$analyze(result)
print(defuse3$summary(analysis))

## End(Not run)
## For full examples, try:
## browseURL(system.file("full_doc/defuse3.html", package="ASSISTant"))
```

---

generateDiscreteData *A data generation function using a discrete distribution for Rankin score rather than a normal distribution*

Description

A data generation function using a discrete distribution for Rankin score rather than a normal distribution

Usage

```
generateDiscreteData(prevalence, N, support = 0L:6L, ctlDist, trtDist)
```

Arguments

`prevalence`	a vector of group prevalences (length denoted by J below)
`N`	the sample size to generate
`support`	the support values of the discrete distribution (length K), default 0:6
`ctlDist`	a probability vector of length K denoting the Rankin score distribution for control.
`trtDist`	an K x J probability matrix with each column is the Rankin distribution for the associated group

Value

a three-column data frame of `subGroup`, `trt` (0 or 1), and `score`

Examples

```
# Simulate data from a discrete distribution for the Rankin scores,
# which are typically ordinal integers from 0 to 6 in the following
# simulations. So we define a few scenarios.
library(ASSISTant)
null.uniform <- rep(1, 7L) ## uniform on 7 support points
hourglass <- c(1, 2, 2, 1, 2, 2, 1)
inverted.hourglass <- c(2, 1, 1, 2, 1, 1, 2)
bottom.heavy <- c(2, 2, 2, 1, 1, 1, 1)
bottom.heavier <- c(3, 3, 2, 2, 1, 1, 1)
top.heavy <- c(1, 1, 1, 1, 2, 2, 2)
top.heavier <- c(1, 1, 1, 2, 2, 3, 3)
ctlDist <- null.uniform
trtDist <- cbind(null.uniform, null.uniform, hourglass, hourglass) ## 4 groups
generateDiscreteData(prevalence = rep(1, 4), N = 10, ctlDist = ctlDist,
                     trtDist = trtDist) ## default support is 0:6
trtDist <- cbind(bottom.heavy, bottom.heavy, top.heavy, top.heavy)
generateDiscreteData(prevalence = rep(1, 4), N = 10, ctlDist = ctlDist,
                     trtDist = trtDist)
support <- c(-2, -1, 0, 1, 2) ## Support of distribution
top.loaded <- c(1, 1, 1, 3, 3) ## Top is heavier
ctl.dist <- c(1, 1, 1, 1, 1) ## null on 5 support points
trt.dist <- cbind(ctl.dist, ctl.dist, top.loaded) ## 3 groups
generateDiscreteData(prevalence = rep(1, 3), N = 10, support = support,
                     ctlDist = ctl.dist, trtDist = trt.dist)
```

generateNormalData	*A data generation function along the lines of what was used in the Lai, Lavori, Liao paper. score rather than a normal distribution*

Description

A data generation function along the lines of what was used in the Lai, Lavori, Liao paper. score rather than a normal distribution

Usage

```
generateNormalData(prevalence, N, mean, sd)
```

Arguments

`prevalence`	a vector of group prevalences (length denoted by J below)
`N`	the sample size to generate
`mean`	a 2 x J matrix of means under the null (first row) and alternative for each group
`sd`	a 2 x J matrix of standard deviations under the null (first row) and alternative for each group

Value

a three-column data frame of `subGroup`, `trt` (0 or 1), and `score`

groupSampleSize	*Compute the sample size for any group at a stage assuming a nested structure as in the paper.*

Description

In the three stage design under consideration, the groups are nested with assumed prevalences and fixed total sample size at each stage. This function returns the sample size for a specified group at a given stage, where the futility stage for the overall group test may be specified along with the chosen subgroup.

Usage

```
groupSampleSize(prevalence, N, stage, group, HJFutileAtStage = NA,
  chosenGroup = NA)
```

Arguments

`prevalence`	the vector of prevalence, will be normalized if not already so. The length of this vector implicitly indicates the number of groups J.
`N`	an integer vector of length 3 indicating total sample size at each of the three stages
`stage`	the stage of the trial
`group`	the group whose sample size is desired
`HJFutileAtStage`	is the stage at which overall futility occured. Default `NA` indicating it did not occur. Also ignored if stage is 1.
`chosenGroup`	the selected group if HJFutilityAtStage is not `NA`. Ignored if stage is 1.

Value

the sample size for group

References

Adaptive Choice of Patient Subgroup for Comparing Two Treatments by Tze Leung Lai and Philip W. Lavori and Olivia Yueh-Wen Liao. Contemporary Clinical Trials, Vol. 39, No. 2, pp 191-200 (2014). http://www.sciencedirect.com/science/article/pii/S1551714414001311

`LLL.SETTINGS` *Design and trial settings used in the Lai, Lavori, Liao paper simulations*

Description

A list of design and trial design settings used for analysis and simulations in the Lai, Lavori, Liao paper displayed in Tables 1 and 2. The elements of the list are the following

trialParameters **N** the sample size at each of three interim looks, the last being the final one; The length of this also determines the number of interim looks

- **type1Error** the overall type I error
- **eps** the fraction of type I error spent at each interim look
- **type2Error** the type II error desired

scenarios A list of the 10 settings used in the simulations named `S0`, `S1`, ..., `S10` as in the paper, each with three elements

- **mean** a $2 \times J$ matrix of means, the first row for the null setting, the second for the alternative
- **sd** a $2 \times J$ matrix of standard deviations, the first row for the null setting, the second for the alternative

prevalences A list of two elements with prevalence vectors used in the paper; the lengths of these vectors implicitly define the number of groups.

- **table1** a vector of equal prevalences for six groups used in table 1
- **table2** a vector of prevalences used in table 2 of the paper

References

Adaptive Choice of Patient Subgroup for Comparing Two Treatments by Tze Leung Lai and Philip W. Lavori and Olivia Yueh-Wen Liao. Contemporary Clinical Trials, Vol. 39, No. 2, pp 191-200 (2014). `http://www.sciencedirect.com/science/article/pii/S1551714414001311`

`mHP.b` *Compute the efficacy boundary (modified Haybittle-Peto) for the first two stages*

Description

Compute the efficacy boundary (modified Haybittle-Peto) for the first two stages

Usage

```
mHP.b(prevalence, N, cov.J, mu.prime, Sigma.prime, alpha, btilde, theta)
```

Arguments

`prevalence` the vector of prevalences between 0 and 1 summing to 1. J, the number of groups, is implicitly the length of this vector and should be at least 2.

`N` a three-vector of total sample size at each stage

`cov.J` the 3 x 3 covariance matrix for Z_J at each of the three stages

`mu.prime` a list of J mean vectors, each of length $J-1$ representing the conditional means of all the other Z_j given Z_i. This mean does not account for the conditioned value of Z_i and so has to be multiplied by that during use!

`Sigma.prime` a list of J covariance matrices, each $J-1$ by $J-1$ representing the conditional covariances all the other Z_j given Z_i

`alpha` the amount of type I error to spend

`btilde` the futility boundary

`theta` the effect size on the probability scale

References

Adaptive Choice of Patient Subgroup for Comparing Two Treatments by Tze Leung Lai and Philip W. Lavori and Olivia Yueh-Wen Liao. Contemporary Clinical Trials, Vol. 39, No. 2, pp 191-200 (2014). `http://www.sciencedirect.com/science/article/pii/S1551714414001311`

`mHP.btilde` *Compute the futility boundary (modified Haybittle-Peto) for the first two stages*

Description

The futility boundary $\tilde{b}$ is computed by solving (under the alternative)

Usage

```
mHP.btilde(beta, cov.J)
```

Arguments

`beta`	the type II error
`cov.J`	the 3 x 3 covariance matrix

Details

$$P(\tilde{Z}_J^1 \leq \tilde{b} or \tilde{Z}_J^2 \leq \tilde{b}) = \epsilon\beta$$

where the superscripts denote the stage and ϵ is the fraction of the type I error (α) spent and β is the type II error. We make use of the joint normal density of Z_J (the overall group) at each of the three stages and the fact that the $\tilde{Z}_J$ is merely a translation of Z_J. So here the calculation is based on a mean of zero and has to be translated during use!

References

Adaptive Choice of Patient Subgroup for Comparing Two Treatments by Tze Leung Lai and Philip W. Lavori and Olivia Yueh-Wen Liao. Contemporary Clinical Trials, Vol. 39, No. 2, pp 191-200 (2014). http://www.sciencedirect.com/science/article/pii/S1551714414001311

`mHP.c` *Compute the efficacy boundary (modified Haybittle-Peto) for the final (third) stage*

Description

Compute the efficacy boundary (modified Haybittle-Peto) for the final (third) stage

Usage

```
mHP.c(prevalence, N, cov.J, mu.prime, Sigma.prime, alpha, btilde, b, theta)
```

Arguments

`prevalence` the vector of prevalences between 0 and 1 summing to 1. J, the number of groups, is implicitly the length of this vector and should be at least 2.

`N` a three-vector of total sample size at each stage

`cov.J` the 3 x 3 covariance matrix for Z_J at each of the three stages

`mu.prime` a list of J mean vectors, each of length $J-1$ representing the conditional means of all the other Z_j given Z_i. This mean does not account for the conditioned value of Z_i and so has to be multiplied by that during use!

`Sigma.prime` a list of J covariance matrices, each $J-1$ by $J-1$ representing the conditional covariances all the other Z_j given Z_i

`alpha` the amount of type I error to spend

`btilde` the futility boundary

`b` the efficacy boundary for the first two stages

`theta` the effect size on the probability scale

References

Adaptive Choice of Patient Subgroup for Comparing Two Treatments by Tze Leung Lai and Philip W. Lavori and Olivia Yueh-Wen Liao. Contemporary Clinical Trials, Vol. 39, No. 2, pp 191-200 (2014). `http://www.sciencedirect.com/science/article/pii/S1551714414001311`

`wilcoxon` *Compute the standardized Wilcoxon test statistic for two samples*

Description

We compute the standardized Wilcoxon test statistic with mean 0 and and standard deviation 1 for samples x and y. The R function `stats::wilcox.test()` returns the statistic

Usage

```
wilcoxon(x, y, theta = 0)
```

Arguments

`x` a sample numeric vector

`y` a sample numeric vector

`theta` a value > 0 but < 1/2.

Details

$$U = \sum_i R_i - \frac{m(m+1)}{2}$$

where R_i are the ranks of the first sample x of size m. We compute

$$\frac{(U - mn(1/2 + \theta))}{\sqrt{mn(m+n+1)/12}}$$

where θ is the alternative hypothesis shift on the probability scale, i.e. $P(X > Y) = 1/2 + \theta$.

Value

the standardized Wilcoxon statistic

Index