Package ‘CLAST’

May 7, 2026

Type Package

Title Exact Confidence Limits after a Sequential Trial

Version 1.0.1

Date 2022-04-02

Author Chris J. Lloyd

Maintainer Chris J. Lloyd <`c.lloyd@mbs.edu`>

Description

The user first provides design vectors n, a and b as well as null (p0) and alternative (p1) benchmark values for the probability of success. The key function ``mv.plots.SM()'' calculates mean values of exact upper and lower limits based on four different rank ordering methods. These plots form the basis of selecting a rank ordering. The function ``inference()'' calculates exact limits from a provided realisation and ordering choice. For more information, see ``Exact confidence limits after a group sequential single arm binary trial'' by Lloyd, C.J. (2020), Statistics in Medicine, Volume 38, 2389-2399, <doi:10.1002/sim.8909>.

License GPL-2

Depends R(>= 3.2)

Imports stats, graphics

Encoding UTF-8

RoxygenNote 7.1.2

NeedsCompilation no

Repository CRAN

Date/Publication 2022-04-08 07:02:40 UTC

Contents

`CLAST-package` *Confidence Limits After Sequential Trial Exact Confidence Limits after a Sequential Trial*

Description

The user first provides design vectors n, a and b as well as null (p0) and alternative (p1) benchmark values for the probability of success. The key function "mv.plots.SM()" calculates mean values of exact upper and lower limits based on four different rank ordering methods. These plots form the basis of selecting a rank ordering. The function "inference()" calculates exact limits from a provided realisation and ordering choice. For more information, see "Exact confidence limits after a group sequential single arm binary trial" by Lloyd, C.J. (2020), Statistics in Medicine, Volume 38, 2389-2399, <doi:10.1002/sim.8909>.

Details

This package allows the user to compare different methods of calculating exact upper and lower limits for a probability after a group sequential trial.

Author(s)

Chris J. Lloyd Maintainer: Chris J. Lloyd <c.lloyd@mbs.edu>

References

Lloyd, C.J. (2020) Exact confidence limits after a group sequential single arm binary trial. Statistics in Medicine, Volume 38, 2389-2399. doi: 10.1002/sim.8909

Examples

```
# Example 1 in table 1 of Lloyd (2020)
  n=c(5,6,5,9)
  a=c(2,4,5,12)
  b=c(5,9,11,13)
  plt.sample.space.SM(n,a,b) # Produces figure 1 in reference
#
  p0=.4
  p1=.75
  errors.SM(n,a,b,p0,p1) # Gives type 1 and type 2 errors
  plt.sample.space.SM(n,a,b,p0,p1) # Add error rates into plot title
#
# Selection of type of limits is based on the diagnostic plots
# from the next command, which produces three panel graphic
# identical to Figure 2 in the main reference. The results here
# support method LR
  mv.plots.SM(n,a,b,p0=p0,p1=p1)
# Once we have an actual outcome we can calculate the limits.
  y=c(4,2,5) # Trial terminates on trial 2 since total successes is 11.
  inference(n,a,b,y,type="LR")
# Produces exact limits for specific outcome y.
```

`CP.lower` *Calculates Clopper-Pearson lower limit*

Description

Calculates the exact Clopper-Pearson lower limit for a binomial probability based on x successes from n trials

Usage

```
CP.lower(x, n, a = 0.05)
```

Arguments

`x`	number of successes (integer between 0 and n inclusive)
`n`	number of binary trials
`a`	coverage error of limit

Value

A numeric value between 0 and 1

Author(s)

Chris J. Lloyd

References

Clopper, C. and Pearson, E. S. (1934). The use of confidence or fiducial limits illustrated in the case of the binomial. Biometrika, 26 (4): 404-413. doi: 10.2307/2331986

Examples

```
# Find exact 99% lower limit for probability based on 20 successes from 55 trials.
CP.lower(20,55,a=0.01)
# Find exact (95%) lower limit for probability based on all failures.
CP.lower(0,5)
```

`CP.stats.SM` *Calculates all possible Clopper-Pearson limits.*

Description

Calculates all possible Clopper-Pearson limits for p from an object that contains all possible data sets from a group sequential trial.

Usage

```
CP.stats.SM(obj, alpha = 0.05, type = "upper")
```

Arguments

`obj`	a list with elements S, M and design, containing all possible values of total successes S and stopping stage M based on the given design. Typically this is the output of function sample.space or sample.space.2 and will also have elements Y and decision.
`alpha`	coverage error of exact limits
`type`	character, either "upper" or "lower"

Value

a list with element "lims" as well as all elements of argument obj. The extra element contains the CP limit for each possible outcome.

Author(s)

Chris J. LLoyd

References

Clopper, C. and Pearson, E. S. (1934). The use of confidence or fiducial limits illustrated in the case of the binomial. Biometrika, 26 (4): 404-413. doi: 10.2307/2331986

Examples

```
n=c(5,6,5,9)
a=c(2,4,5,12)
b=c(5,9,11,13)
# There are 364 possible outcomes from this design which are
# listed in a natural systematic order by function sample.space.
all.samples=sample.space(n,a,b)
attributes(all.samples)
# Y contains the 364 possible sequential binary outcomes;
# M contains how many stages before the decision;
# S contains the total number of success that produces the decision;
# decision the final binary test result of H0 or H1.
CP.stats.SM(obj=all.samples)
```

`CP.upper` *Calculates Clopper-Pearson upper limit*

Description

Calculates the exact Clopper-Pearson upper limit for a binomial probability based on x successes from n trials

Usage

```
CP.upper(x, n, a = 0.05)
```

Arguments

`x`	number of successes (integer between 0 and n inclusive)
`n`	number of binary trials
`a`	coverage error of limit

Value

A numeric value between 0 and 1

Author(s)

Chris J. Lloyd

References

Clopper, C. and Pearson, E. S. (1934). The use of confidence or fiducial limits illustrated in the case of the binomial. Biometrika, 26 (4): 404-413. doi: 10.2307/2331986

Examples

```
# Find exact 99% upper limit for probability based on 20 successes from 55 trials.
CP.upper(20,55,a=0.01)
# Find exact (95%) upper limit for probability from all successes.
CP.lower(5,5)
```

`cross` *Special combination of Matrix and Vector*

Description

This operation is useful in recursively generating all the possible path histories of a sequential design.

Usage

```
cross(A, v)
```

Arguments

`A`	An arbitrary matrix
`v`	An arbitrary vector

Details

For each component of v, the matrix A is duplicated with an extra right column equal to v[i]. This is added to the current output until the components of v are exhausted.

Value

A matrix with dim(A)[2]+1 columns and dim(A)[1]*length(v) rows.

Author(s)

Chris J. Lloyd

Examples

```
A=cbind(c(1,2),c(3,4))
v=c(5,6)
cross(A,v)
#     [,1] [,2] [,3]
#[1,]    1    3    5
#[2,]    2    4    5
#[3,]    1    3    6
#[4,]    2    4    6
```

`errors.SM` *Exact error rates of specified sequential design*

Description

For a given sequential design defined by vectors a, b and n, calculate the probability of a type 1 error (assuming p=p0) and type 2 error (assuming p=p1)

Usage

```
errors.SM(n, a, b, p0, p1)
```

Arguments

`n`	Design vector of planned sample sizes
`a`	Design vector of lower futility boundaries
`b`	Design vector of upper superiority boundaries
`p0`	Lower benchnmark for success probability
`p1`	Upper benchnmark for success probability

Value

list with attributes type1 and type 2

Author(s)

Chris J. Lloyd

Examples

```
# Example 1 in table 1 of Lloyd (2020)
n=c(5,6,5,9)
a=c(2,4,5,12)
b=c(5,9,11,13)
p0=.4
p1=.75
errors.SM(n,a,b,p0,p1)
# $type1
# [1] 0.09590162
# $type2
# [1] 0.1060701
```

`exact.lower.limits.SM` *Calculates all exact lower limits.*

Description

Calculates Buehler lower limit for all possible data sets based on a provided ranking function.

Usage

```
exact.lower.limits.SM(obj, lims = NULL, alpha = 0.05, set = FALSE)
```

Arguments

`obj`	list with components $count, $S, $N, typically the output of LR.stats.SM, CP.stats.SM, JT.rank.SM or ML.rank.SM. In this case, it will also have element $lims.
`lims`	a vector of same length as obj$count, most often an approximate lower limits. If this is missing, then it should be an element of obj$lims.
`alpha`	exact coverage error
`set`	If true, exact limits of 1 are replaced by the largest limit less than 1. This will never be chosen by the user and is there for certain numerical investigations.

Value

A numeric vector of same length as obj$lims, giving the exact lower limit for each possible outcome as listed in components obj$M and obj$S.

Author(s)

Chris J. Lloyd

References

Lloyd, C.J. (2020) Exact confidence limits after a group sequential single arm binary trial. Statistics in Medicine, Volume 38, 2389-2399.

Examples

```
# Example 1 in table 1 of Lloyd (2020)
n=c(5,6,5,9)
a=c(2,4,5,12)
b=c(5,9,11,13)
# There are 364 possible outcomes from this design which are
# listed in a natural systematic order by function sample.space.
all.samples=sample.space.SM(n,a,b)
attributes(all.samples)
# Y contains the 364 possible sequential binary outcomes;
# M contains how many stages before the decision;
# S contains the total number of success that produces the decision;
```

```
# decision the final binary test result of H0 or H1.
all.CP=CP.stats.SM(all.samples,type="lower")
all.CP$lims # These limits are based on fixed sample size so are not edxact.
all.exact=exact.lower.limits.SM(all.CP)
plot(all.CP$lims,all.exact)
```

`exact.upper.limits.SM` *Calculates all exact upper limits.*

Description

Calculates Buehler upper limit for all possible data sets based on ranking function provided in obj$lims.

Usage

```
exact.upper.limits.SM(obj, lims = NULL, alpha = 0.05, set = FALSE)
```

Arguments

`obj`	list with components $count, $S, $N, typically the output of LR.stats.SM, CP.stats.SM, JT.rank.SM or ML.rank.SM. In this case, it will also have element $lims.
`lims`	a vector of same length as obj$count, most often an approximate lower limits. If this is missing, then it should be an element of obj$lims.
`alpha`	exact coverage error
`set`	If true, exact limits of 0 are replaced by the smallest limit greater than 1. This will never be chosen by the user and is there for certain numerical investigations.

Value

A numeric vector of same length as obj$lims, giving the exact lower limit for each possible outcome as listed in components obj$M and obj$S.

Author(s)

Chris J. Lloyd

References

Lloyd, C.J. (2020) Exact confidence limits after a group sequential single arm binary trial. Statistics in Medicine, Volume 38, 2389-2399.

Examples

```
# Example 1 in table 1 of Lloyd (2020)
n=c(5,6,5,9)
a=c(2,4,5,12)
b=c(5,9,11,13)
# There are 364 possible outcomes from this design which are
# listed in a natural systematic order by function sample.space.
all.samples=sample.space.SM(n,a,b)
attributes(all.samples)
# Y contains the 364 possible sequential binary outcomes;
# M contains how many stages before the decision;
# S contains the total number of success that produces the decision;
# decision the final binary test result of H0 or H1.
all.CP=CP.stats.SM(all.samples,type="upper")
all.CP$lims # These limits are based on fixed sample size so are not edxact.
all.exact=exact.upper.limits.SM(all.CP)
plot(all.CP$lims,all.exact)
```

`inference` *Exact limits from outcome*

Description

Calculates exact upper and lower confidence limits from the outcome of a multi-stage group sequential trial.

Usage

```
inference(n, a, b, y, alpha, type)
```

Arguments

`n`	Design vector of planned sample sizes
`a`	Design vector of lower futility boundaries
`b`	Design vector of upper superiority boundaries
`y`	Actual outcome of experiment
`alpha`	exact coverage error
`type`	character, either "LR", "CP", "JT" or "ML"

Value

list with components

`lower`	exact lower limit, type dependent
`upper`	exact lower limit, type dependent
`est`	ML estimate
`type`	selected ordering function "LR", "CP", "JT" or "ML"

Author(s)

Chris J. Lloyd

References

Lloyd, C.J. (2020) Exact confidence limits after a group sequential single arm binary trial. Statistics in Medicine, Volume 38, 2389-2399.

Examples

```
# Example from table 1 of Lloyd (2020)
  n=c(5,6,5,9)
  a=c(2,4,5,12)
  b=c(5,9,11,13)
  y=c(4,2,5) # This is the actual outcome. The trial stopped for
#              superiority at stage m=3 with s=11 successes.
  inference(n,a,b,y,type="CP")
# Results happen to be identical for type="LR" but different
# for the alternative ordering functions "JT" and "ML"
```

`JT.rank.SM` *Calculates Jennison & Turnbull ranking of sample space*

Description

Calculates the Jennison & Turnbull (1983) ranking of each possible outcome of the sample space generated from a sequential design.

Usage

```
JT.rank.SM(obj)
```

Arguments

`obj`	a list with elements S, M and design, containing all possible values of total successes S and stopping stage M based on the given design. Typically this is the output of function sample.space or sample.space.2 and will also have elements Y and decision.

Value

a list with element "lims" as well as all elements of argument obj. The extra element contains the JT rank of each possible the outcome.

Author(s)

Chris J. Lloyd

References

Jennison C, Turnbull BW. Confidence intervals for a binomial parameter following a multistage test with application to MIL-STD 105D and medical trials. Technometrics. 1983(25), 49-58. doi: 10.1080/00401706.1983.10487819

Examples

```
n=c(5,6,5,9)
a=c(2,4,5,12)
b=c(5,9,11,13)
# There are 364 possible outcomes from this design which are
# listed in a natural systematic order by function sample.space.
all.samples=sample.space(n,a,b)
attributes(all.samples)
# Y contains the 364 possible sequential binary outcomes;
# M contains how many stages before the decision;
# S contains the total number of success that produces the decision;
# decision the final binary test result of H0 or H1.
JT.rank.SM(obj=all.samples)
# Component "lims" contains the rank of each sequential binary
# outcome. The same rank is assigned to outcomes with the same
# values of (S,M).
```

`LR.lower` *Calculates likelihood ratio based lower limit*

Description

Calculates approximate lower limit for a probability from x successes out of n binary trials based on the signed root likelihood ratio.

Usage

```
LR.lower(x, n, a = 0.05, epsilon = 1e-18)
```

Arguments

`x`	number of successes (integer between 0 and n inclusive)
`n`	number of binary trials
`a`	coverage error of limit
`epsilon`	Tolerance supplied to uniroot.

Value

A numeric values between 0 and 1

Author(s)

Chris J. Lloyd

Examples

```
# Find approximate 99% lower limit for probability based on 20 successes from 55 trials.
LR.lower(20,55,a=0.01)
# Find exact (95%) lower limit for probability based on all failures.
LR.lower(0,5)
```

`LR.stats.SM` *Calculates all possible LR limits.*

Description

Calculates all possible LR based limits for p from an object that contains all possible data sets from a group sequential trial.

Usage

```
LR.stats.SM(obj, alpha = 0.05, type = "upper")
```

Arguments

`obj`	a list with elements S, M and design, containing all possible values of total successes S and stopping stage M based on the given design. Typically this is the output of function sample.space or sample.space.2 and will also have elements Y and decision.
`alpha`	nominal coverage error of approximate LR based limits
`type`	character, either "upper" or "lower"

Value

a list with element "lims" as well as all elements of argument obj. The extra element contains the LR exact limit for each possible outcome.

Author(s)

Chris J. Lloyd

Examples

```
n=c(5,6,5,9)
a=c(2,4,5,12)
b=c(5,9,11,13)
# There are 364 possible outcomes from this design which are
# listed in a natural systematic order by function sample.space.
all.samples=sample.space(n,a,b)
attributes(all.samples)
# Y contains the 364 possible sequential binary outcomes;
# M contains how many stages before the decision;
# S contains the total number of success that produces the decision;
# decision the final binary test result of H0 or H1.
LR.stats.SM(obj=all.samples)
```

`LR.upper` *Calculates likelihood ratio based upper limit*

Description

Calculates approximate upper limit for a probability from x successes out of n binary trials based on the signed root likelihood ratio.

Usage

```
LR.upper(x, n, a = 0.05, epsilon = 1e-18)
```

Arguments

`x`	number of successes (integer between 0 and n inclusive)
`n`	number of binary trials
`a`	coverage error of limit
`epsilon`	Tolerance supplied to uniroot.

Value

A numeric values between 0 and 1

Author(s)

Chris J. Lloyd

Examples

```
# Find approximate 99% upper limit for probability based on 20 successes from 55 trials.
LR.upper(20,55,a=0.01)
# Find exact (95%) upper limit for probability from all successes.
LR.upper(5,5)
```

`ML.rank.SM` *Maximum likelihood estimator of p.*

Description

Calculates the ML estimator for each possible outcome of the sample space generated from a sequential design.

Usage

```
ML.rank.SM(obj)
```

Arguments

`obj`	a list with elements S, M and design, containing all possible values of total successes S and stopping stage M based on the given design. Typically this is the output of function sample.space or sample.space.2 and will also have elements Y and decision.

Value

a list with element "lims" as well as all elements of argument obj. The extra element contains the ML estimator of p for each possible outcome.

Author(s)

Chris J. Lloyd

Examples

```
n=c(5,6,5,9)
a=c(2,4,5,12)
b=c(5,9,11,13)
# There are 364 possible outcomes from this design which are
# listed in a natural systematic order by function sample.space.
all.samples=sample.space(n,a,b)
attributes(all.samples)
# Y contains the 364 possible sequential binary outcomes;
# M contains how many stages before the decision;
# S contains the total number of success that produces the decision;
# decision the final binary test result of H0 or H1.
ML.rank.SM(obj=all.samples)$lims
# Component "lims" contains the value of the ML estimate i.e.
# total successes S divided by total trials N. The same ML
# estimator occurs for outcomes with the same values of (S,M).
```

`mv.plots.SM` *Diagnostic mean values plots.*

Description

Plots mean value of upper limit, lower limit and interval width for four different ranking methods. This function is basically a wrapper for mv.plot.

Usage

```
mv.plots.SM(n, a, b, type = "interval",
B = 100, offset = TRUE, plt = c(1, 1, 1), p0 = NULL, p1 = NULL, focus = FALSE)
```

Arguments

`n`	Design vector of planned sample sizes
`a`	Design vector of lower futility boundaries
`b`	Design vector of upper superiority boundaries
`type`	Either "upper", "lower" or "interval" (default)
`B`	Integer controlling fineness of plot (default=100)
`offset`	if TRUE then ML mean value is subtracted
`plt`	Logical vector indicating output plots of upper, lower and interval (default=c(1,1,1))
`p0`	Lower (null) benchmark for success probability
`p1`	Upper (alternative) benchmark for success probability
`focus`	Logical. If true, plots are restricted to p between p0 and p1. (default=FALSE)

Value

NULL

Author(s)

Chris J. Lloyd

References

Lloyd, C.J. (2021) Exact confidence limits after a group sequential single arm binary trial. Statistics in Medicine, Volume 38, 2389-2399. doi: 10.1002/sim.8909

Examples

```
# Figure 2 in Lloyd (2020)
  n=c(5,6,5,9)
  a=c(2,4,5,12)
  b=c(5,9,11,13)
  p0=.4
  p1=.75
  mv.plots.SM(n,a,b,p0=p0,p1=p1)
# Produces three panel graphic identical to Figure 2 in reference
  mv.plots.SM(n,a,b,p0=p0,p1=p1,focus=TRUE)
# Produces alternative graphic focussed on relevant values of p.
# In both cases LR (in blue) appears best. CP can perform poorly
# for values of p outside the range of interest.
```

`mv.SM` *Mean value of upper limits.*

Description

Calculates mean value of a provided vector of upper limits as a function of p

Usage

```
mv.SM(obj, lims = NULL, p = NULL, B = 99, offset = TRUE, wgt = TRUE)
```

Arguments

`obj`	list with components $S, $N, $count and optionally $lims, typically the output of sample.space.SM
`lims`	if not a component of obj
`p`	vector of values of p at which to calculate mean value
`B`	number of evenly spaced values of p if not provided
`offset`	if TRUE then ML mean value is subtracted
`wgt`	if TRUE than assign zero probability weight to extreme limits of 1 or 0.

Value

list with elements $x (containing grid of B values of probability) and $y (containing corresponding mean values)

Author(s)

Chris J. Lloyd

Examples

```
n=c(5,6,5,9)
a=c(2,4,5,12)
b=c(5,9,11,13)
# Enumerate all possible elements of the sufficiency reduced samples
# space i.e. all values of S and M. Also listed are the counts and
# subcounts of these outcomes, the test decision and the vector
# n, a, and b in $design

data.SM=sample.space.SM(n,a,b) # There are 26 elements.

# Calculate all approximate LR upper limits for these 26 outcomes.
all.LR.high=LR.stats.SM(data.SM,type="upper")$lims
# Calculate all approximate LR upper limits for these 26 outcomes.
all.LR.low=LR.stats.SM(data.SM,type="lower")$lims

# Calculate the mean values of these lims as a function of p
mv.high=mv.SM(data.SM,all.LR.high,p=(1:99)/100,offset=FALSE)
mv.low=mv.SM(data.SM,all.LR.low,p=(1:99)/100,offset=FALSE)
plot(c(0,1),c(0,1),ylab="mean value",xlab="p",type="n")
lines(mv.high)
lines(mv.low)
abline(0,1,lty=3)
title(main="Mean value of upper and lower limits by p")
```

`plt.sample.space.SM` *Sample space for given sequential design.*

Description

Plots decision function in S-M space with boundary vectors "a" and "b". If p0 and p1 are provided then the type 1 and type 2 error are displayed in the main graphic title

Usage

```
plt.sample.space.SM(n, a, b, p0 = NULL, p1 = NULL)
```

Arguments

`n`	Design vector of sample sizes
`a`	Design vector of lower boundaries
`b`	Design vector of upper boundaries
`p0`	Lower benchnmark for success probability
`p1`	Upper benchnmark for success probability

Value

NULL

Author(s)

Chris J. Lloyd

References

Lloyd, C.J. (2020) Exact confidence limits after a group sequential single arm binary trial. Statistics in Medicine, Volume 38, 2389-2399. doi: 10.1002/sim.8909

Examples

```
# Figure 1 of Lloyd (2020)
  n=c(5,6,5,9)
  a=c(2,4,5,12)
  b=c(5,9,11,13)
  plt.sample.space.SM(n,a,b)
# produces plots in Figure 1 or reference.
```

`prob.SM` *Probability of sufficient statistics (S,M).*

Description

Calculates probability distribution of sufficient statistics (S,M) from sample space object.

Usage

```
prob.SM(data, p, m = NULL, s = NULL)
```

Arguments

`data`	list with components $Y, $S, $M, $design, $count. Thsi will typically be the output of sample.space or sample.space.2.
`p`	value of binary probability
`m`	number of stages at end of sequential trial
`s`	number of successes at end of sequential trial

Value

list with component $prob, $count, $subcount and matrix $data giving the $subcount different binary sequential outcomes that lead to $prob.

Author(s)

Chris J. Lloyd

References

Lloyd, C.J. (2020) Exact confidence limits after a group sequential single arm binary trial. Statistics in Medicine, Volume 38, 2389-2399. doi: 10.1002/sim.8909

Examples

```
n=c(5,6,5,9)
a=c(2,4,5,12)
b=c(5,9,11,13)
# There are 364 possible outcomes from this design which are
# listed in a natural systematic order by function sample.space.
all.samples=sample.space(n,a,b)
attributes(all.samples)
# Y contains the 364 possible sequential binary outcomes;
# M contains how many stages before the decision;
# S contains the total number of success that produces the decision;
# decision the final binary test result of H0 or H1.
prob.SM(all.samples,p=.5,m=3,s=11)
```

`RANK` *Produces ranks of entries of vector*

Description

An alternative to the function rank (which replaces ties by average rank). This function instead gives each tied value the same rank. The output will have the same length as x but contain integer values from 1 to length(unique(x)).

Usage

```
RANK(x)
```

Arguments

`x`	a numeric vector

Value

Vector of integer ranks of the same length as x.

Author(s)

Chris J. Lloyd

Examples

```
x=c(5,4,4,7,6)
rank(x)
# 3.0 1.5 1.5 5.0 4.0
RANK(x)
# 2 1 1 4 3
```

`sample.space` *Sample space enumeration (K>2)*

Description

Creates all possible samples from a multi-stage group sequential trial with K>2. If K=2, use the function sample.space.2 instead.

Usage

```
sample.space(n, a, b)
```

Arguments

`n`	Design vector of planned sample sizes
`a`	Design vector of lower futility boundaries
`b`	Design vector of upper superiority boundaries

Value

list with components $Y, $M, $S, $decision and $design

Author(s)

Chris J. Lloyd

References

Lloyd, C.J. (2020) Exact confidence limits after a group sequential single arm binary trial. Statistics in Medicine, Volume 38, 2389-2399. doi: 10.1002/sim.8909

Examples

```
n=c(5,6,5,9)
a=c(2,4,5,12)
b=c(5,9,11,13)
# There are 364 possible outcomes from this design which are
# listed in a natural systematic order by function sample.space.
all.samples=sample.space(n,a,b)
attributes(all.samples)
# Y contains the 364 possible sequential binary outcomes;
```

```
# M contains how many stages before the decision;
# S contains the total number of success that produces the decision;
# decision the final binary test result of H0 or H1.
```

`sample.space.2` *Sample space enumeration (K=2)*

Description

Creates all possible samples from a multi-stage group sequential trial with K=2. If K>2, use the function sample.space instead.

Usage

```
sample.space.2(n, a, b)
```

Arguments

`n`	Design vector of planned sample sizes
`a`	Design vector of lower futility boundaries
`b`	Design vector of upper superiority boundaries

Value

list with components $Y, $M, $S, $decision and $design

Author(s)

Chris J. Lloyd

References

Lloyd, C.J. (2020) Exact confidence limits after a group sequential single arm binary trial. Statistics in Medicine, Volume 38, 2389-2399. doi: 10.1002/sim.8909

Examples

```
n=c(5,6)
a=c(2,6)
b=c(5,7)
# There are 18 possible outcomes from this design which are
# listed in a natural systematic order by function sample.space.
all.samples=sample.space.2(n,a,b)
attributes(all.samples)
# Y contains the 18 possible sequential binary outcomes;
# M contains how many stages before the decision;
# S contains the total number of success that produces the decision;
# decision the final binary test result of H0 or H1.
```

`sample.space.SM` *Sample space enumeration*

Description

Creates all possible samples from a multi-stage group sequential trial for K>=2 by calling sample.space or sample.space.2.

Usage

```
sample.space.SM(n,a,b)
```

Arguments

`n`	Design vector of planned sample sizes
`a`	Design vector of lower futility boundaries
`b`	Design vector of upper superiority boundaries

Value

list with components

`M`	number of trials for each possible trial outcome
`S`	number of responses for each possible trial outcome
`N`	number of patients for each possible trial outcome
`count`	combinatoric multiplier for probability distribution
`subcountM`	combinatoric components for probability distribution
`decision`	test decision for each possible trial outcome
`design`	list giving design vectors

Author(s)

Chris J. Lloyd

References

Lloyd, C.J. (2020) Exact confidence limits after a group sequential single arm binary trial. Statistics in Medicine, Volume 38, 2389-2399. doi: 10.1002/sim.8909

Examples

```
# Example 1 in table 1 of Lloyd (2020)
  n=c(5,6,5,9)
  a=c(2,4,5,12)
  b=c(5,9,11,13)
  sample.space.SM(n,a,b)
# Object describing all 26 possible outcomes for (s,m).
```

Index