Package ‘gsrsb’

June 25, 2019

Type Package
Title Group Sequential Refined Secondary Boundary
Version 1.1.1
Date 2019-06-24
Author Jiangtao Gou and Fengqing (Zoe) Zhang
Maintainer Jiangtao Gou <gourpackage@gmail.com>
Description A gate-keeping procedure to test a primary and a secondary endpoint in a group sequential design with multiple interim looks. Computations related to group sequential primary and secondary boundaries. Refined secondary boundaries are calculated for a gatekeeping test on a primary and a secondary endpoint in a group sequential design with multiple interim looks. The choices include both the standard boundaries and the boundaries using error spending functions. Version 1.0.0 was released on April 12, 2017. See Tamhane et al. (2018), “A gatekeeping procedure to test a primary and a secondary endpoint in a group sequential design with multiple interim looks”, Biometrics, 74(1), 40-48.

License GPL-3
Encoding UTF-8
LazyData true
RoxygenNote 6.1.1.9000
Imports stats, mvtnorm, ldbounds, xtable
NeedsCompilation no
Repository CRAN
Date/Publication 2019-06-25 13:50:03 UTC

R topics documented:

  cdBoundary ......................................................... 2
genCorrMat .......................................................... 3
initLocPeak .......................................................... 4
ldldInitLocPeak ....................................................... 5
ldNominalSig ........................................................... 7
Description

Generate lower and upper bounds for programs calculating the secondary endpoint’s type I error when the correlation rho between the primary endpoint and the secondary endpoint equals 1.

Usage

cdBoundary(cvec, dvec, gammaVec, dlt, upper = TRUE)

Arguments

cvec primary boundary.
dvec secondary boundary.
gammaVec square root of information vector.
dlt test statistic of the primary endpoint follows a normal distribution with mean dlt and standard deviation 1.
upper type of bounds, upper bound is TRUE, lower bound is FALSE.

Details

This function generates upper and lower bounds for further computation. For more details, refer to Tamhane et al. (2018, Biometrics), section 4.2.

Value

lower and upper bounds for programs calculating the secondary endpoint’s type I error when the correlation rho is 1.
**genCorrMat**

**Author(s)**
Jiangtao Gou

**References**

**Examples**

```r
cvec <- rep(1.992,3)
dvec <- c(1.535*sqrt(3),1.535*sqrt(3/2),1.535)
gammaVec <- c(sqrt(1/3),sqrt(2/3),1)
dlt <- 2
ubBoundary <- cdBoundary(cvec, dvec, gammaVec, dlt, upper=TRUE)
```

---

**genCorrMat**

*Correlation Matrix Generator*

**Description**
Generate correlation matrix between standardized sample mean test statistics for the two endpoint at different looks.

**Usage**

```r
genCorrMat(gammaVec, type, rhoPS = 0)
```

**Arguments**

- `gammaVec`: a vector which contains `gamma_(1), ..., gamma_(K-1), gamma_(K)`, square root of information vector.
- `type`: type of primary or secondary endpoint. For primary endpoint calculation, `type` is 1, the returned matrix is K by K. For secondary endpoint calculation, `type` is 2, the returned matrix is (K+1) by (K+1).
- `rhoPS`: correlation between primary and secondary endpoints.

**Details**
This function generates correlation matrix between different mean statistics. For more details, refer to Tamhane et al. (2018, Biometrics), section 2.

**Value**
correlation matrix, K by K for primary endpoint, (K+1) by (K+1) for secondary endpoint, where K is the number of interims.
initLocPeak

Author(s)

Jiangtao Gou
Fengqing (Zoe) Zhang

References


Examples

corrMat <- genCorrMat(gammaVec=c(sqrt(1/3),sqrt(2/3),1), type=2, rhoPS = 0.3)

initLocPeak

*Find the Location of Maximum, Standard OBF and POC*

Description

Calculate the location of maximal type I error of the standard O’Brien-Fleming and Pocock refined secondary boundaries.

Usage

`initLocPeak(alpha, tVec, cvec, type = 2, initIntvl = c(1, 4))`

Arguments

- `alpha`: type I error.
- `tVec`: information vector.
- `cvec`: primary group sequential boundary.
- `type`: type of the test procedure for the secondary endpoint. O’Brien-Fleming (OBF) type error spending function is 1, Pocock (POC) type error spending function is 2.
- `initIntvl`: computing parameter, a pair of numbers containing the end-points of the interval to be searched for the root.

Details

This function search the location of the maximal point, in order to calculate the standard (original) O’Brien-Fleming (OBF) and Pocock (POC) refined secondary boundaries.

Value

location of maximum, a number between 1 and the number of interims
**Author(s)**

Jiangtao Gou

**References**


**See Also**

SecondaryBoundary, ldInitLocBeak

**Examples**

```r
#require(mvtnorm)
#K <- 8
#gammaVec <- sqrt((1:K)/K)
#tVec <- gammaVec*2
#alpha = 0.025
#c <- 2.072274
#cvec <- c/gammaVec
#loc <- initLocPeak(alpha, tVec, cvec, type=2, initIntvl=c(1,3))
```

---

**ldInitLocPeak**

*Find the Location of Maximum, Error Spending Approach*

**Description**

Calculate the location of maximal type I error of secondary endpoint.

**Usage**

```r
ldInitLocPeak(alpha, tVec, cvec, type = 2, initIntvl = c(0.8, 4))
```

**Arguments**

- `alpha` type I error.
- `tVec` information vector.
- `cvec` primary group sequential boundary.
ldInitLocPeak

**type**

Type of the test procedure for the secondary endpoint. O’Brien- Fleming (OBF) type error spending function is 1, Pocock (POC) type error spending function is 2.

**initIntvl**

Computing parameter, a pair of numbers containing the end-points of the interval to be searched for the root.

**Details**

This function searches the location of maximal type I error of secondary endpoint by using the error spending approach.

**Value**

Location of maximum, a number between 1 and the number of interims.

**Author(s)**

Jiangtao Gou

**References**


**See Also**

ldSecondaryBoundary, initLocBeak

**Examples**

```r
# require(mvtnorm)
# require(ldbounds)
# K <- 6;
# tVec <- c(140, 328, 453, 578, 659, 1080)/1080;
# alpha = 0.025;
# cvec.obf <- bounds(tVec, iuse=c(1), alpha=c(alpha));
# cvec <- cvec.obf$upper.bounds;
# loc <- ldInitLocPeak(alpha, tVec, cvec, type=2, initIntvl=c(0.9, 4))
```
Description

Nominal significance for the secondary endpoint are calculated by using the error spending approach.

Usage

```r
ldNominalSig(alpha, tVec, cvec, locPeak, type = 2, initIntvl = c(1, 4))
```

Arguments

- **alpha**: original significance level.
- **tVec**: information vector.
- **cvec**: primary group sequential boundary.
- **locPeak**: location of maximum, a number between 1 and the number of interims.
- **type**: O’Brien- Fleming (OBF) type error spending function is 1, Pocock (POC) type error spending function is 2.
- **initIntvl**: computing parameter, a pair of numbers containing the end-points of the interval to be searched for the root.

Details

This function calculates the nominal significance level of any Lan-DeMets error spending boundary. The original significance level is used to choose the initial searching range of the nominal significance.

Value

nominal significance of the secondary group sequential boundary.

Author(s)

Jiangtao Gou

References


ldPrimaryBoundary

Calculate Primary Boundaries, the Error Spending Approach

Description

Primary boundaries calculation of Lan-DeMets OBF and POC.

Usage

ldPrimaryBoundary(tVec, alpha, type = 1, initIntvl = c(0.8, 8))

Arguments

tVec a vector of information, gammaVec = sqrt(tVec).
alpha significance level
type type of sequential procedure. OBF is 1, POC is 2.
initIntvl parameter for function uniroot (two numbers)

Value

a vector of primary boundaries.

Author(s)

Jiangtao Gou

References


IdSecControl

**Description**

Calculate the difference between the error rate and significance level for the secondary endpoint, Lan-DeMets error spending approach.

**Usage**

```r
IdSecControl(ap, alpha, cvec, tVec, ExtrimLoc, type = 2)
```

**Arguments**

- `ap`: significance level for the primary endpoint
- `alpha`: targeted significance level for the secondary endpoint
- `cvec`: a vector of calculated primary boundaries
- `tVec`: a vector of information, `gammaVec = sqrt(tVec)`
- `ExtrimLoc`: an integer between 1 and K, locate the maximum of type I error of secondary endpoint
- `type`: type of sequential procedures. Type 1 OBF d, Type 2 POC d.

**Value**

difference between alpha and the calculated error rate.

**Author(s)**

Jiangtao Gou

**References**


**See Also**

secControl
**ldSecondaryBoundary**  
*Calculate Refined Secondary Boundary, Error Spending Approach*

**Description**

Refined secondary boundaries are calculated by using the error spending approach.

**Usage**

```r
ldSecondaryBoundary(alpha, tVec, cvec, locPeak, type = 2,
                      initIntvl = c(0.6, 4))
```

**Arguments**

- `alpha`: original significance level.
- `tVec`: information vector.
- `cvec`: primary group sequential boundary.
- `locPeak`: location of maximum, a number between 1 and the number of interims.
- `type`: type of the test procedure for the secondary endpoint. O'Brien-Fleming (OBF) type error spending function is 1, Pocock (POC) type error spending function is 2.
- `initIntvl`: computing parameter, a pair of numbers containing the end-points of the interval to be searched for the root.

**Details**

This function calculates the refined secondary boundaries of any Lan-DeMets error spending boundary based on the primary boundaries.

**Value**

refined secondary boundaries.

**Author(s)**

Jiangtao Gou

**References**


nominalSig

See Also

secondaryBoundary, secondaryBoundaryVecLD

Examples

```r
# require(mvtnorm)
# require(lldbounds)
# K <- 6;
# tVec <- c(140, 328, 453, 578, 659, 1080)/1080;
# alpha = 0.025;
# cvec.obf <- bounds(tVec, iuse=c(1), alpha=c(alpha));
# cvec <- cvec.obf$upper.bound;
# secbound <- ldSecondaryBoundary(alpha, tVec, cvec, locPeak=4, type=2, #     initIntvl=c(0.8, 8))
```

nominalSig  

Calculate Nominal Significance, Standard Approach

Description

Nominal significance for the secondary endpoint are calculated by using the standard (original) approach.

Usage

nominalSig(gammaVec, cvec)

Arguments

gammaVec  
square root of information.

cvec  
group sequential boundary.

Details

This function calculates the nominal significance level of any given boundary.

Value

nominal significance

Author(s)

Jiangtao Gou
References


See Also

ldNominalSig, secondaryBoundaryVecOrig

Examples

```r
# require(mvtnorm)
# require(ldbounds)
# nSig <- nominalSig(gammaVec=c(sqrt(1/3),1),cvec=c(2.2,1.8))
```

---

**primaryBoundary**  
*Calculate Primary Boundaries, Standard Approach*

**Description**

Primary boundaries calculation of standard (original) OBF and POC.

**Usage**

```r
primaryBoundary(gammaVec, alpha, type = 1, initIntvl = c(1, 4))
```

**Arguments**

- `gammaVec` a vector of square root of information.
- `alpha` significance level
- `type` type of sequential procedure. OBF is 1, POC is 2.
- `initIntvl` parameter for function uniroot (two numbers)

**Value**

original OBF and POC boundaries (primary endpoints) (a number, c(K)).

**Author(s)**

Jiangtao Gou
References

See Also
ldprimaryboundary

primaryBoundaryVec  Calculate the Primary Boundaries

Description
Primary boundaries are calculated, including the standard approach and the error spending approach.

Usage

primaryBoundaryVec(alpha, tVec, OBF = TRUE, LanDeMets = FALSE, digits = 2, printOut = TRUE, initIntvl = c(1, 8))

Arguments

alpha  significance level for the primary endpoint.
tVec  information (vector).
OBF  type of procedures. TRUE for OBF, FALSE for POC.
LanDeMets  type of procedures. TRUE for Lan-Demets type boundaries, FALSE for original boundaries.
digits  number of digits for output,
printOut  TRUE for printing the boundaries.
initIntvl  parameter for function uniroot (two numbers) for function primaryBoundary or function ldPrimaryBoundary

Value

OBF and POC boundaries (primary endpoints) (vector).

Author(s)
Jiangtao Gou
References


Examples

```r
# require(mvtnorm)
# K = 4
# alpha = 0.025
# tVec = (1:K)/K
# boundaryVector <- primaryBoundaryVec(alpha, tVec, initIntvl=c(1,4),
#   OBFR=TRUE, LanDemets=FALSE, digits=3, printOut=TRUE)
# boundaryVector <- primaryBoundaryVec(alpha, tVec, initIntvl=c(1,4),
#   OBFR=FALSE, LanDemets=FALSE, digits=3, printOut=TRUE)
# boundaryVector <- primaryBoundaryVec(alpha, tVec, initIntvl=c(1,8),
#   OBFR=TRUE, LanDemets=TRUE, digits=3, printOut=TRUE)
# boundaryVector <- primaryBoundaryVec(alpha, tVec, initIntvl=c(1,4),
#   OBFR=FALSE, LanDemets=TRUE, digits=3, printOut=TRUE)
```

---

psbTeXtable

**Summarize Primary and Refined Secondary Boundaries in a TeX table**

Description

Primary boundaries and refined secondary boundaries are listed in a TeX table.

Usage

```r
psbTeXtable(alpha, tVec, pOBF = TRUE, sOBF = FALSE,
             LanDemets = FALSE, digits = 2)
```

Arguments

- `alpha`: type I error probability.
- `tVec`: vector of relative information levels. The last element in the vector is 1.
\texttt{psbTeXtable} type of boundary, \texttt{TRUE} is the O’Brien-Fleming boundary, \texttt{FALSE} is the
Pocock boundary.

\texttt{sOBF} type of boundary, \texttt{TRUE} is the O’Brien-Fleming boundary, \texttt{FALSE} is the
Pocock boundary.

\texttt{LanDeMets} type of boundary, \texttt{TRUE} is the error spending approach, \texttt{FALSE} is the
original approach.

\texttt{digits} number of digits after decimal point to display in the table.

**Details**

This function gives a TeX format table including both primary boundary and refined secondary boundary. The number of digits after decimal point can be specified through parameter \texttt{digits}.

**Value**

a TeX format table including both primary boundary and refined secondary boundary.

**Author(s)**

Jiangtao Gou

Fengqing (Zoe) Zhang

**References**

Glimm, E., Maurer, W., and Bretz, F. (2010). Hierarchical testing of multiple endpoints in group-

endpoints in group sequential or adaptive clinical trials. \textit{Journal of Biopharmaceutical Statistics} 17,
1201-1210.


70, 659-663.

35, 549-556.

64, 191-199.

group sequential design. \textit{Biometrics} 66, 1174-1184.

to test a primary and a secondary endpoint in a group sequential design with multiple interim looks.
Examples

```r
#require(mvtnorm)
#require(lbounds)
#require(xtable)
#psbTextable(alpha=0.025,tVec=c(1/2,3/4,1),pOBF=TRUE,sOBF=FALSE,LanDeMets=FALSE)
```

---

**refinedBoundary**

*Summarize Primary and Refined Secondary Boundaries, Nominal Significance*

**Description**

Primary boundaries, refined secondary boundaries, and nominal significance for the secondary endpoint are listed.

**Usage**

```r
refinedBoundary(alpha, tVec, pOBF = TRUE, sOBF = FALSE,
                 LanDeMets = FALSE, digits = 2)
```

**Arguments**

- `alpha`: type I error probability.
- `tVec`: vector of relative information levels. The last element in the vector is 1.
- `pOBF`: type of primary boundary, `TRUE` is the O’Brien-Fleming boundary, `FALSE` is the Pocock boundary.
- `sOBF`: type of secondary boundary, `TRUE` is the O’Brien-Fleming boundary, `FALSE` is the Pocock boundary.
- `LanDeMets`: type of boundary, `TRUE` is the error spending approach, `FALSE` is the original approach.
- `digits`: number of digits after decimal point for primary and secondary boundaries.

**Details**

This function gives a list including primary boundary, refined secondary boundary, and the nominal significance for the secondary endpoint. The number of digits for the nominal significance depends on parameter `alpha`.

**Value**

A result list including primary boundary, refined secondary boundary, and the nominal significance for the secondary endpoint.

**Author(s)**

Jiangtao Gou
References


Examples

```r
require(mvtnorm)
require(ldbounds)
result <- refinedBoundary(alpha=0.05, tVec=c(0.2,0.6,1))
result$primaryBoundary
result$secondaryBoundary
result$nominalSignificance
```

<table>
<thead>
<tr>
<th>secControl</th>
<th>Difference between the Error Rate and Significance Level, Standard Approach</th>
</tr>
</thead>
</table>

Description

Calculate the difference between the error rate and significance level for the secondary endpoint, standard (original) approach.

Usage

```r
secControl(d, alpha, cvec, gammaVec, ExtrmLoc, type = 2)
```
**Arguments**

- **d**: boundary of secondary endpoint at the final look (a number, \(d_{(K)}\))
- **alpha**: targeted significance level for the secondary endpoint
- **cvec**: a vector of calculated primary boundaries
- **gammaVec**: square root of information
- **extrmloc**: an integer between 1 and K, locate the maximum of type I error of secondary endpoint
- **type**: type of sequential procedures. Type 1 OBF \(d\), Type 2 POC \(d\).

**Value**

difference between alpha and the calculated error rate.

**Author(s)**

Jiangtao Gou

**References**


**See Also**

ldSecControl

---

**Description**

Calculate the standard O’Brien-Fleming and Pocock refined secondary boundaries

**Usage**

```r
secondaryBoundary(alpha, tVec, cvec, locPeak, type = 2,
      initIntvl = c(1, 4))
```
Arguments

- **alpha**: type I error.
- **tVec**: information vector.
- **cvec**: primary group sequential boundary.
- **locPeak**: location of maximum, a number between 1 and the number of interims.
- **type**: type of the test procedure for the secondary endpoint. O’Brien- Fleming (OBF) type error spending function is 1, Pocock (POC) type error spending function is 2.
- **initIntvl**: computing parameter, a pair of numbers containing the end-points of the interval to be searched for the root.

Details

This function calculates the standard (original) O’Brien-Fleming (OBF) and Pocock (POC) refined secondary boundaries.

Value

standard O’Brien-Fleming and Pocock refined secondary boundaries.

Author(s)

Jiangtao Gou

References


See Also

`ldSecondaryBoundary`, `initLocBeak`

Examples

```r
#require(mvtnorm)
#K <- 8
#gammaVec <- sqrt((1:K)/K)
#tVec <- gammaVec^2
#alpha = 0.025
#c <- 2.072274
#cvec <- c/gammaVec
#loc <- initLocPeak(alpha,tVec,cvec,type=2,initIntvl=c(1,4))
```
Calculate Refined Secondary Boundaries and Nominal Significance

**Description**

Refined secondary boundaries, and nominal significance for the secondary endpoint are calculated.

**Usage**

```r
secondaryBoundaryVec(alpha, tVec, pOBF = TRUE, sOBF = FALSE, LanDeMets = FALSE, initIntvl = c(0.8, 8))
```

**Arguments**

- `alpha` type I error probability.
- `tVec` vector of relative information levels. The last element in the vector is 1.
- `pOBF` type of primary boundary, `TRUE` is the O’Brien-Fleming boundary, `FALSE` is the Pocock boundary.
- `sOBF` type of secondary boundary, `TRUE` is the O’Brien-Fleming boundary, `FALSE` is the Pocock boundary.
- `LanDeMets` type of boundary, `TRUE` is the error spending approach, `FALSE` is the original approach.
- `initIntvl` computing parameter, a pair of numbers containing the end-points of the interval to be searched for the root.

**Details**

This function gives a list including refined secondary boundary and the nominal significance for the secondary endpoint. There are a computing parameter `initIntvl`. Parameter `initIntvl` contains the end-points of the interval to be searched for the root. For Lan-DeMets error spending approach, the lower end point should choose a number slightly less than 1, and the upper end point should choose a number between 4 and 10.

**Value**

- a result list including refined secondary boundary and the nominal significance for the secondary endpoint.

**Author(s)**

Jiangtao Gou
References


See Also

`secondaryBoundaryVecLD`, `secondaryBoundaryVecOrig`

Examples

```r
# require(mvtnorm)
# require(lcbounds)
# result <- secondaryBoundaryVec(alpha=0.025,tVec=c(1/2,1),pOBF=TRUE,sOBF=FALSE,
# LanDeMets=FALSE,initIntvl=c(0.8,5))
# result$secondaryBoundary
# result$nominalSignificance
```

---

**secondaryBoundaryVecLD**

*Calculate Refined Secondary Boundaries and Nominal Significance, the Error Spending Approach*

**Description**

Lan-DeMets refined secondary boundaries, and nominal significance for the secondary endpoint are calculated by using the error spending approach.
secondaryBoundaryVecLD

Usage

secondaryBoundaryVecLD(alpha, tVec, primaryOBF = TRUE, secondaryOBF = FALSE, initIntvl = c(0.8, 8))

Arguments

alpha  type I error probability.
tVec  vector of relative information levels. The last element in the vector is 1.
primaryOBF  type of primary boundary, TRUE is the O'Brien-Fleming boundary, FALSE is the Pocock boundary.
secondaryOBF  type of secondary boundary, TRUE is the O'Brien-Fleming boundary, FALSE is the Pocock boundary.
initIntvl  computing parameter, a pair of numbers containing the end-points of the interval to be searched for the root.

Details

This function uses the Lan-DeMets error spending approach, and gives a list including refined secondary boundary and the nominal significance for the secondary endpoint. There is a computing parameter initIntvl. Parameter initIntvl contains the end-points of the interval to be searched for the root. For Lan-DeMets error spending approach, the lower end point should choose a number slightly less than 1, and the upper end point should choose a number between 4 and 10.

Value

a result list including Lan-DeMets refined secondary boundary and the nominal significance for the secondary endpoint.

Author(s)

Jiangtao Gou

References


secondaryBoundaryVecOrig


**See Also**

`secondaryBoundaryVec`, `secondaryBoundaryVecOrig`

**Examples**

```r
# require(mvtnorm)
# require(ldbounds)
# result <- secondaryBoundaryVecLD(alpha=0.025, tVec=c(1/2, 1), primaryOBF=TRUE,
#   secondaryOBF=FALSE, initIntvl=c(0.8, 6))
# result$secondaryBoundary
# result$nominalSignificance
```

### secondaryBoundaryVecOrig

*Calculate Refined Secondary Boundaries and Nominal Significance, Standard Approach*

**Description**

Standard refined secondary boundaries, and nominal significance for the secondary endpoint are calculated by using the standard (original) approach.

**Usage**

```r
secondaryBoundaryVecOrig(alpha, tVec, primaryOBF = TRUE,
  secondaryOBF = FALSE, initIntvl = c(1, 8))
```

**Arguments**

- `alpha`  
  type I error probability.

- `tVec`  
  vector of relative information levels. The last element in the vector is 1.

- `primaryOBF`  
  type of primary boundary, TRUE is the O’Brien-Fleming boundary, FALSE is the Pocock boundary.

- `secondaryOBF`  
  type of secondary boundary, TRUE is the O’Brien-Fleming boundary, FALSE is the Pocock boundary.

- `initIntvl`  
  computing parameter, a pair of numbers containing the end-points of the interval to be searched for the root.
Details

This function uses the standard approach (O’Brien and Fleming 1979, Pocock 1977), and gives a list including refined secondary boundary and the nominal significance for the secondary endpoint. There is a computing parameter initIntvl. Parameter initIntvl contains the end-points of the interval to be searched for the root. The lower end point should choose a number around 1, and the upper end point should choose a number between 4 and 10.

Value

a result list including standard refined secondary boundary and the nominal significance for the secondary endpoint.

Author(s)

Jiangtao Gou

References


See Also

`secondaryBoundaryVec`, `secondaryBoundaryVecLD`

Examples

```r
#require(mvtnorm)
#require(lbounds)
#result <- secondaryBoundaryVecOrig(alpha=0.025,tVec=c(1/2,1),primaryOBF=TRUE,
#    secondaryOBF=FALSE, initIntvl=c(1,4))
#result$secondaryBoundary
#result$nomialSignificance
```
Index

cdBoundary, 2

genCorrMat, 3

initLocPeak, 4

ldInitLocPeak, 5
ldNominalSig, 7
ldPrimaryBoundary, 8
ldSecControl, 9
ldSecondaryBoundary, 10

nominalSig, 11

primaryBoundary, 12
primaryBoundaryVec, 13
psbTeXtable, 14

refinedBoundary, 16

secControl, 17
secondaryBoundary, 18
secondaryBoundaryVec, 20
secondaryBoundaryVecLD, 21
secondaryBoundaryVecOrig, 23