Package: MultiGroupSequential (via r-universe)

October 30, 2024

Type Package

Version 1.0.0

Date 2022-03-31

Title Group-Sequential Procedures with Multiple Endpoints

Description Provides various testing procedures for group-sequential trials with multiple endpoints. Two sets of procedures are provided.

Depends R (>= 4.0.0)

Imports stats, rpact, gMCP, mvtnorm, OpenMx, hommel, minidown

License GPL (>= 2)

RoxygenNote 7.1.2

LazyData true

Suggests knitr, rmarkdown

VignetteBuilder knitr

Repository https://marvels2031.r-universe.dev

RemoteUrl https://github.com/marvels2031/multigroupsequential

RemoteRef HEAD

RemoteSha 0c9e830c268337fdaf57f76910ef96f4d0647a20

Contents

MultiGroupSequential-package 2
crosslist 2
hxhochberg 3
hxhommel 4
xccalgsp 5
xccalgspcor 6
xccalgspsim 7
xccalgspsim1 8

xccrit 9
xcmaurerbretz 10
xcseqgxgs 11
xcseqhh 12
xcseqhhgs 13
xcspending 14

Index **16**

`MultiGroupSequential-package` *Group-Sequential Procedures with Multiple Endpoints*

Description

Provides various testing procedures for group-sequential trials with multiple endpoints. Two sets of procedures are provided.

Details

The DESCRIPTION file: This package was not yet installed at build time.

Index: This package was not yet installed at build time.

Author(s)

Xiaodong Luo [aut, cre], Hui Quan [ctb], Sanofi [cph]

Maintainer: Xiaodong Luo <Xiaodong.Luo@sanofi.com>

References

Luo (2022)

`crosslist` *Cross list all the scenarios*

Description

Cross list all the scenarios

Usage

```
crosslist(b=list(a1=c(2,3),a2=c(2,4),a3=c(0,1)))
```

Arguments

`b`	A list of lists

Value

`df`	dataframe consisting of all the scenarios row by row

Author(s)

Xiaodong Luo

Examples

```
crosslist(b=list(pbo.hazard=c(0.1,0.15,0.3),
                 hazard.ratio=c(0.6,0.7,0.8),
                 censoring.rate=c(0.05,0.06,0.07)
                 )
         )
```

`hxhochberg` *Hochberg procedure with different alphas for different endpoints*

Description

Hochberg procedure with different alphas for different endpoints

Usage

```
hxhochberg(pvalues,alpha,epsilon=1.0e-10,precision=10)
```

Arguments

`pvalues`	p-values from different endpoints
`alpha`	same length as `pvalues` with (different) alphas for different endpoints
`epsilon`	lower bound for the alpha
`precision`	precision of the values

Details

This procedure handles Hochberg procedure with different alphas for different endpoints

Value

`decisions`	an index of rejected hypoetheses

Author(s)

Xiaodong Luo

Examples

```
hxhochberg(pvalues=runif(5),alpha=seq(0.01,0.025,len=5),epsilon=1.0e-10,precision=10)
```

`hxhommel` *Hommel procedure with different alphas for different endpoints*

Description

Hommel procedure with different alphas for different endpoints

Usage

```
hxhommel(pvalues,alpha,epsilon=1.0e-10,precision=10)
```

Arguments

`pvalues`	p-values from different endpoints
`alpha`	same length as `pvalues` with (different) alphas for different endpoints
`epsilon`	lower bound for the alpha
`precision`	precision of the values, obsolete for backward compatibility

Details

It turns out [hommel](https://cran.r-project.org/web/packages/hommel/) can handle Hommel procedure with different alpha's for different endpoints, the function 'hxhommel' is just a wrapper function.

Value

`decisions`	an index of rejected hypotheses

Author(s)

Xiaodong Luo

Examples

```
hxhommel(pvalues=runif(5),alpha=seq(0.01,0.025,len=5),epsilon=1.0e-10,precision=10)
```

xccalgsp *Calculate group-sequential p-values*

Description

This will calculate group-sequential p-values

Usage

```
xccalgsp(xm=qnorm(matrix(rep(c(0.03,0.04,0.01),times=2),ncol=3,nrow=2)),
                 alpham=matrix(rep(c(0.02,0.03,0.05),each=2),ncol=3,nrow=2),
           informationm=matrix(rep(c(0.4,0.8,1),each=2),ncol=3,nrow=2),direction=-1)
```

Arguments

`xm` a matrix of test statistics for each endpoint (in row) and each interim (in column)

`alpham` a matrix of alpha spending for the statistics `xm`

`informationm` a matrix of information fractions for the statistics `xm`

`direction` -1: (one-sided)reject if test stat is smaller than or equal to the critical value; 1: (one-sided)reject if test stat is greater than or equal to the critical value; 0: (two-sided)reject if the absolute value of the test stat is greater than the critical value

Value

`pm` group-sequential p-values

`critm` critical values

Author(s)

Xiaodong Luo

Examples

```
xm=qnorm(matrix(rep(c(0.03,0.04,0.01),times=4),ncol=3,nrow=4))
im=matrix(rep(c(0.4,0.8,1),each=4),ncol=3,nrow=4)
alpham=matrix(0,nrow=4,ncol=3)
for (i in 1:4){
   alpham[i,]=xcspending(alpha=0.025,fractions=im[i,],family="OBF",rho=(i+1)/2)$aseq
}
xccalgsp(xm=xm,alpham=alpham,informationm=im,direction=-1)
```

xccalgspcor *Calculate group-sequential p-values given the correlation matrix of the test statistics*

Description

This will calculate group-sequential p-values given the correlation matrix of the test statistics

Usage

```
xccalgspcor(xm=qnorm(c(0.03,0.04,0.01)),
                    alpham=c(0.02,0.03,0.05),
                    corrm=diag(length(xm)),direction=-1,tol=1e-10)
```

Arguments

`xm`	a vector of test statistics at each analysis
`alpham`	a vector of alpha spending for the statistics `xm`
`corrm`	correlation matrix of the statistics `xm`
`direction`	-1: (one-sided)reject if test stat is smaller than or equal to the critical value; 1: (one-sided)reject if test stat is greater than or equal to the critical value; 0: (two-sided)reject if the absolute value of the test stat is greater than the critical value
`tol`	accuracy tolerance when calculating the quantiles

Value

`pm`	group-sequential p-values
`critm`	critical values

Author(s)

Xiaodong Luo

Examples

```
xm=qnorm(matrix(rep(c(0.03,0.04,0.01),times=2),ncol=3,nrow=2))
ir=c(0.4,0.8,1)
corrm=diag(length(ir))
for (i in 1:length(ir))for(j in 1:length(ir))corrm[i,j]=sqrt(ir[pmin(i,j)]/ir[pmax(i,j)])
xccalgsp(xm=xm)$critm[1,]
xccalgspsim(xm=xm)$critm[1,]
xccalgspcor(xm=xm[1,],corrm=corrm)$critm
```

`xccalgspsim` *Calculate group-sequential p-values via simulation*

Description

This will calculate group-sequential p-values via simulation

Usage

```
xccalgspsim(xm=qnorm(matrix(rep(c(0.03,0.04,0.01),times=2),ncol=3,nrow=2)),
                      alpham=matrix(rep(c(0.02,0.03,0.05),each=2),ncol=3,nrow=2),
                      informationm=matrix(rep(c(0.4,0.8,1),each=2),ncol=3,nrow=2),
                      r.seed=rep(17,2),nsample=1e+6,direction=-1)
```

Arguments

`xm`	a matrix of test statistics for each endpoint (in row) and each interim (in column)
`alpham`	a matrix of alpha spending for the statistics `xm`
`informationm`	a matrix of information fractions for the statistics `xm`
`r.seed`	random seeds for each endpoints
`nsample`	number of random samples
`direction`	-1: (one-sided)reject if test stat is smaller than or equal to the critical value; 1: (one-sided)reject if test stat is greater than or equal to the critical value; 0: (two-sided)reject if the absolute value of the test stat is greater than the critical value

Value

`pm`	group-sequential p-values
`critm`	critical values

Note

This provides the calculation for the group-sequential p-values in case there is an issue in using rpact package.

Author(s)

Xiaodong Luo

Examples

```
xm=qnorm(matrix(rep(c(0.03,0.04,0.01),times=4),ncol=3,nrow=4))
im=matrix(rep(c(0.4,0.8,1),each=4),ncol=3,nrow=4)
alpham=matrix(0,nrow=4,ncol=3)
for (i in 1:4){
   alpham[i,]=xcspending(alpha=0.025,fractions=im[i,],family="OBF",rho=(i+1)/2)$aseq
}
xccalgspsim(xm=xm,alpham=alpham,informationm=im,r.seed=rep(17,4),direction=-1)
```

xccalgspsim1 *Calculate group-sequential p-values via simulation*

Description

This utility function will be called by function "xccalgspsim" to calculate group-sequential p-values via simulation for single endpoint

Usage

```
xccalgspsim1(xm=qnorm(c(0.03,0.04,0.01)),alpham=c(0.02,0.03,0.05),
                        informationm=c(0.4,0.8,1),
                        r.seed=17,nsample=1e+6,direction=0)
```

Arguments

xm	test statistics
alpham	alpha spending
informationm	information fractions
r.seed	random seed
nsample	number of random samples
direction	-1: (one-sided)reject if test stat is smaller than or equal to the critical value; 1: (one-sided)reject if test stat is greater than or equal to the critical value; 0: (two-sided)reject if the absolute value of the test stat is greater than the critical value

Value

crit.value	critical values
p.value.gs	group-sequential p-values
xm	test statistics
alpham	alpha spending
informationm	information fractions

Note

This provides the calculation for the group-sequential p-values in case there is an issue in using rpact package.

Author(s)

Xiaodong Luo

Examples

```
xm=qnorm(c(0.03,0.04,0.01))
im=c(0.4,0.8,1)
alpham=xcspending(alpha=0.025,fractions=im,family="OBF",rho=2)$aseq
xccalgspsim1(xm=xm,alpham=alpham,informationm=im,direction=-1)
```

---

xccrit *Calculate critical values*

Description

This utility function calculates the critical values

Usage

```
xccrit(direction=-1,alpha=0.025,informationRates=c(0.4,0.7,1),
                 userAlphaSpending=c(0.01,0.015,0.025),alpha.low=1e-10)
```

Arguments

`direction`	-1: (one-sided)reject if test stat is smaller than or equal to the critical value; 1: (one-sided)reject if test stat is greater than or equal to the critical value; 0: (two-sided)reject if the absolute value of the test stat is greater than the critical value
`alpha`	overall familywise error rate
`informationRates`	information fractions
`userAlphaSpending`	alpha spent at each interim
`alpha.low`	default is 1e-10, if allocated alpha is smaller than this number, the corresponding critical value will be set to infinity

Value

`crit`	critical values

Author(s)

Xiaodong Luo

Examples

```
xccrit(direction=-1,alpha=0.025,informationRates=c(0.4,0.7,1),
               userAlphaSpending=c(0.01,0.015,0.025),alpha.low=1e-10)
```

`xcmaurerbretz` *Maurer-Bretz sequential graphical approach*

Description

This will conduct group-sequential testing for multiple endpoints based on Maurer-Bretz approach

Usage

```
xcmaurerbretz(xm=qnorm(matrix(rep(c(0.03,0.04,0.01),times=4),ncol=3,nrow=4)),
                   informationm=matrix(rep(c(0.4,0.8,1),each=4),ncol=3,nrow=4),
                         spending=rep("OBF",4),param.spending=rep(1,4),
                     alpha=0.025,direction=-1,graphin=BonferroniHolm(nrow(xm)),
                         alpha.low=1e-10,retrospective=0)
```

Arguments

`xm` a matrix of test statistics for each endpoint (in row) and each interim (in column)

`informationm` a matrix of information fractions for the statistics `xm`

`spending` spending functions for each endpoint

`param.spending` parameters in the spending functions

`alpha` overall familywise error rate

`direction` -1: (one-sided)reject if test stat is smaller than or equal to the critical value; 1: (one-sided)reject if test stat is greater than or equal to the critical value; 0: (two-sided)reject if the absolute value of the test stat is greater than the critical value

`graphin` a graph object generated from gMCP

`alpha.low` default is 1e-10, if allocated alpha is smaller than this number, the corresponding critical value will be set to infinity

`retrospective` retrospective: 0 (default) only compares the current test statistic with the updated critical value, 1 compares all the test statistics up to the current one with the updated critical values. Even though retrospective looking at the values is statistically valid in terms of control of the type-1 error rate, no retrospective looking at the past comparisons avoids the dilemma of retrospectively increasing the alpha level for the un-rejected hypothesis in the past

Value

`Hrej` rejected hypotheses

`rejected` the index set of rejected hypotheses

`decisionsm` rejection decision for each endpoint (row) at each timepoint (column)

`cumdecisionsm` cumulative rejection decision for each endpoint (row) at each timepoint (column)

Author(s)

Xiaodong Luo

Examples

```
Sys.setenv(JAVA_HOME="C:/Program Files/Java/jdk-17.0.2/")
library(gMCP)
xm=qnorm(matrix(rep(c(0.03,0.04,0.01),times=4),ncol=3,nrow=4))
im=matrix(rep(c(0.4,0.8,1),each=4),ncol=3,nrow=4)
spending=rep("OBF",4)
param.spending=rep(1,4)
graphin=gMCP::BonferroniHolm(nrow(xm))
xcmaurerbretz(xm=xm,
              informationm=im,
              spending=spending,
              param.spending=param.spending,
              graphin=graphin)
```

`xcseqgxgs` *Sequential graphical procedure for multiple endpoints based on group-sequential p-values*

Description

Sequential graphical procedure for multiple endpoints based on group-sequential p-values

Usage

```
xcseqgxgs(pm=matrix(rep(c(0.03,0.04,0.01),times=2),ncol=3,nrow=2),
          alpha=0.025,graphin=BonferroniHolm(2))
```

Arguments

`pm` a matrix of group-sequential p-values for different endpoints (in row) at different times (in column)

`alpha` overall familywise error rate

`graphin` graph to be used, this is graph object defined by the gMCP package

Value

`rejected` the index set of rejected hypotheses

`decisionsm` rejection decision for each endpoint (row) at each timepoint (column)

`cumdecisionsm` cumulative rejection decision for each endpoint (row) at each timepoint (column)

Note

This provides the calculation for the variance.

Author(s)

Xiaodong Luo

Examples

```
Sys.setenv(JAVA_HOME="C:/Program Files/Java/jdk-17.0.2/")
library(gMCP)
pm=matrix(rep(c(0.03,0.04,0.01),times=2),ncol=3,nrow=2)
graphin=gMCP::BonferroniHolm(2)
xcseqgxgs(pm=pm,alpha=0.025,graphin=graphin)
```

---

xcseqhh *Sequential Hochberg/Hommel procedure for multiple endpoints based on q-values*

Description

Sequential Hochberg/Hommel procedure for multiple endpoints based on q-values

Usage

```
xcseqhh(pm=matrix(rep(c(0.03,0.04,0.01),times=2),ncol=3,nrow=2),
        alpham=matrix(rep(c(0.02,0.03,0.05),each=2),ncol=3,nrow=2),
        epsilon=1.0e-10,precision=10,method='Hochberg')
```

Arguments

`pm` a matrix of group-sequential p-values for different endpoints (in row) at different times (in column)

`alpham` a matrix of alpha spending corresponding to the p-values `pm`

`epsilon` lower bound for the alpha

`precision` precision of the values

`method` "Hochberg" or "Hommel"

Value

`rejected` the index set of rejected hypotheses

`decisionsm` rejection decision for each endpoint (row) at each timepoint (column)

`cumdecisionsm` cumulative rejection decision for each endpoint (row) at each timepoint (column)

`alphaused` alpha levels actually used for each endpoint (row) at each timepoint (column)

Note

This provides the calculation for the variance.

Author(s)

Xiaodong Luo

Examples

```
pm=matrix(rep(c(0.03,0.04,0.01),times=2),ncol=3,nrow=2)
alpham=matrix(rep(c(0.02,0.03,0.05),each=2),ncol=3,nrow=2)
xcseqhh(pm=pm,alpham=alpham,method="Hochberg")
xcseqhh(pm=pm,alpham=alpham,method="Hommel")
```

`xcseqhhgs` *Sequential Hochberg/Hommel procedure for multiple endpoints based on group-sequential p-values*

Description

Sequential Hochberg/Hommel procedure for multiple endpoints based on group-sequential p-values

Usage

```
xcseqhhgs(pm=matrix(rep(c(0.03,0.04,0.01),times=2),ncol=3,nrow=2),
                     alpha=0.025,epsilon=1.0e-10,precision=10,method='Hochberg')
```

Arguments

`pm` a matrix of group-sequential p-values for different endpoints (in row) at different times (in column)

`alpha` overall familywise error rate

`epsilon` lower bound for the alpha

`precision` precision of the values

`method` "Hochberg" or "Hommel"

Value

`rejected` the index set of rejected hypotheses

`decisionsm` rejection decision for each endpoint (row) at each timepoint (column)

`cumdecisionsm` cumulative rejection decision for each endpoint (row) at each timepoint (column)

Note

This provides the calculation for the variance.

Author(s)

Xiaodong Luo

Examples

```
pm=matrix(rep(c(0.03,0.04,0.01),times=2),ncol=3,nrow=2)
xcseqhhgs(pm=pm,alpha=0.025,method="Hochberg")
xcseqhhgs(pm=pm,alpha=0.025,method="Hommel")
```

`xcspending` *Calculate alpha spending*

Description

This utility function calculates alpha spending. Note that the OBF and Pocock spending functions are not the originally proposed ones, they are the modified ones that are closely resemble the original versions. That being said, you might still see some differences

Usage

```
xcspending(alpha,fractions=seq(0.2,1,by=0.2),family="OBF",rho=1)
```

Arguments

`alpha` overall familywise error rate

`fractions` information fractions

`family` family of spending functions, one of "OBF", "pocock", "power"

`rho` parameter of the spending function

Details

OBF: $2\{1-\Phi(\Phi^{-1}(1-\alpha/2)/t^{\rho/2})\}$; pocock: $\alpha\log\{1+(e-1)*t\}$; power: $\alpha * t^{\rho}$

Value

`aseq` alpha spending

Author(s)

Xiaodong Luo

Examples

```
xcspending(alpha=0.025,fractions=seq(0.2,1,by=0.2),family="OBF",rho=1)
```

Index

∗ **Hochberg procedure**
 hxhochberg, 3
∗ **Hochberg**
 xcseqhh, 12
 xcseqhhgs, 13
∗ **Hommel procedure**
 hxhommel, 4
∗ **Hommel**
 xcseqhh, 12
 xcseqhhgs, 13
∗ **Maurer-Bretz**
 xcmaurerbretz, 10
∗ **alpha spending**
 xcspending, 14
∗ **critical values**
 xccrit, 9
 xcspending, 14
∗ **cross listing**
 crosslist, 2
∗ **efficacy boundary**
 xccrit, 9
 xcspending, 14
∗ **graphical procedure**
 xcmaurerbretz, 10
∗ **group-sequential p-values**
 xccalgsp, 5
 xccalgspcor, 6
 xccalgspsim, 7
 xccalgspsim1, 8
 xcseqhhgs, 13
∗ **group-sequential**
 MultiGroupSequential-package, 2
 xccrit, 9
 xcmaurerbretz, 10
 xcseqgxgs, 11
 xcseqhh, 12
 xcseqhhgs, 13
 xcspending, 14
∗ **multiple endpoints**
 MultiGroupSequential-package, 2
∗ **multiplicity control**
 MultiGroupSequential-package, 2
∗ **q-values**
 xcseqhh, 12

crosslist, 2

hxhochberg, 3
hxhommel, 4

MultiGroupSequential
 (MultiGroupSequential-package),
 2
MultiGroupSequential-package, 2

xccalgsp, 5
xccalgspcor, 6
xccalgspsim, 7
xccalgspsim1, 8
xccrit, 9
xcmaurerbretz, 10
xcseqgxgs, 11
xcseqhh, 12
xcseqhhgs, 13
xcspending, 14