Package 'MultiGroupSequential'

August 9, 2023

Type Package

Version 1.1.0

Title Group-Sequential Procedures with Multiple Hypotheses

Description It is often challenging to strongly control the family-wise type-1 error rate in the group-sequential trials with multiple endpoints (hypotheses). The inflation of type-1 error rate comes from two sources (S1) repeated testing individual hypothesis and (S2) simultaneous testing multiple hypotheses. The 'MultiGroupSequential' package is intended to help researchers to tackle this challenge. The procedures provided include the sequential procedures described in Luo and Quan (2023) <doi:10.1080/19466315.2023.2191989> and the graphical procedure proposed by Maurer and Bretz (2013) <doi:10.1080/19466315.2013.807748>. Luo and Quan (2013) describes three procedures, and the functions to implement these procedures are (1) seqgspgx() implements a sequential graphical procedure based on the group-sequential p-values; (2) seqgsphh() implements a sequential Hochberg/Hommel procedure based on the group-sequential p-values; and (3) seqqvalhh() implements a sequential Hochberg/Hommel procedure based on the q-values. In addition, seqmbgx() implements the sequential graphical procedure described in Maurer and Bretz (2013).

Depends R (>= 4.0.0)

Encoding UTF-8

Imports stats, OpenMx, hommel

License GPL (>= 2)

RoxygenNote 7.2.3

NeedsCompilation no

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

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

Repository CRAN

Date/Publication 2023-08-09 14:50:02 UTC

R topics documented:

`calgsp1` *Calculate group-sequential p-values for one hypothesis*

Description

`calgsp1()` calculates the group-sequential p-values for one hypothesis.

Usage

```
calgsp1(
  sx = qnorm(1 - c(0.03, 0.04, 0.01)),
  scrit = qnorm(1 - c(0.01, 0.02, 0.025)),
  salpha = c(0.01, 0.02, 0.025),
  smatrix = diag(3),
  sided = 1
)
```

Arguments

`sx`	Numeric vector of test statistics, assumed to be multivariate normal with variance 1 and correlation matrix given by `smatrix`.
`scrit`	Numeric vector of sequece of critical values for the test statistics in `sx`. It should be computed beforehand. Must have the same length as `sx`.
`salpha`	Numeric vector of cumulative alpha levels for the test statistics in `sx`. Must have the same length as `sx`.
`smatrix`	Matrix with the correlation matrix of the test statistics `sx`.
`sided`	Integer scalar indicating the side of the test:

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

Value

List containing the group-sequential p-values.

Author(s)

Xiaodong Luo

Examples

```
calgsp1(
  sx = qnorm(1 - c(0.03, 0.04, 0.01)),
  scrit = qnorm(1 - c(0.01, 0.02, 0.025)),
  salpha = c(0.01, 0.02, 0.025),
  smatrix = diag(3),
  sided = 1
)
```

---

calgspn *Calculate group-sequential p-values for multiple hypotheses*

Description

`calgspn()` calculates the group-sequential p-values for multiple hypotheses.

Usage

```
calgspn(
  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),
  critm = matrix(rep(qnorm(c(0.02, 0.03, 0.05)), each = 2), ncol = 3, nrow = 2),
  matrix.list = list(diag(3), diag(3)),
  sided = rep(-1, 2)
)
```

Arguments

`xm`	Matrix of test statistics for hypotheses (in row) and each interim (in column).
`alpham`	Matrix of cumulative alpha levels for the test statistics `xm`. Must have the same dimensions as `xm`. For each row, alpha levels must be
`critm`	Matrix of critical values for the test statistics in `xm`. It should be computed beforehand. Must have the same dimensions as `xm`.

`matrix.list` List of correlation matrices corresponding to each hypothesis.

`sided` Integer vector indicating the side of the test:

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

Value

List with element `pm` containing the group-sequential p-values.

Author(s)

Xiaodong Luo

Examples

```
calgspn(
  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),
  critm = matrix(rep(qnorm(c(0.02,0.03,0.05)),each=2),ncol=3,nrow=2),
  matrix.list = list(diag(3),diag(3)),
  sided = rep(-1,2)
)
```

---

checkcrit *Check critical values*

Description

`checkcrit()` is a helper function that checks if the critical values are valid.

Usage

```
checkcrit(
  scrit = qnorm(c(0.01, 0.02, 0.025)),
  salpha = c(0.01, 0.02, 0.025),
  smatrix = diag(3),
  sided = 1
)
```

Arguments

`scrit`	Numeric vector of critical values.
`salpha`	Numeric vector of cumulative alpha levels.
`smatrix`	General correlation matrix.
`sided`	Integer vector indicating the side of the test: • `-1`: Reject if test statistic is smaller than or equal to the critical value (one-sided) • `1`: Reject if test statistic is greater or equal to the critical value (one-sided) • `0`: Reject if the absolute value of the test statistic is greater than the critical value (two-sided)

Value

List with:

- `crit.value`: Critical values
- `salpha`: Cumulative alpha levels passed to `salpha` argument

Author(s)

Xiaodong Luo

Examples

```
checkcrit(
  scrit = qnorm(c(0.01, 0.02, 0.025)),
  salpha = c(0.01, 0.02, 0.025),
  smatrix = diag(3),
  sided = 1
)
```

`findcrit` *Calculate critical values*

Description

`findcirt()` calculates the critical values in the general correlation matrix

Usage

```
findcrit(
  salpha = c(0.01, 0.02, 0.025),
  smatrix = diag(3),
  sided = 1,
  tol = 1e-10,
  alpha.tol = 1e-11
)
```

Arguments

salpha	Numeric vector of cumulative alpha levels.
smatrix	General correlation matrix.
sided	Integer vector indicating the side of the test: • -1: Reject if test statistic is smaller than or equal to the critical value (one-sided) • 1: Reject if test statistic is greater or equal to the critical value (one-sided) • 0: Reject if the absolute value of the test statistic is greater than the critical value (two-sided)
tol	Numeric scalar with the tolerance level for computing critical values.
alpha.tol	Numeric scalar. If the alpha increment is less than this, the critical value is set to a large number determined by alpha.tol.

Value

List with element crit.value containing the obtained critical values.

Author(s)

Xiaodong Luo

Examples

```
findcrit(
  salpha = c(0.01, 0.02, 0.025),
  smatrix = diag(3),
  sided = 1,
  tol = 1e-10,
  alpha.tol = 1e-11
)
```

---

graphical *Graphical procedure*

Description

graphical() performs graphical procedure to test multiple hypotheses

Usage

```
graphical(
  p = c(0.01, 0.04, 0.03),
  W = c(0.5, 0.25, 0.25),
  G = rbind(c(0, 1, 0), c(0, 0, 1), c(1, 0, 0)),
  alpha = 0.05
)
```

Arguments

p	Numeric vector of p-values for the hypotheses.
W	Numeric vector of weigths of the graph. Must have the same length as p.
G	Matrix of the transition matrix of the graph.
alpha	Numeric scalar with the overall type-1 error rate.

Value

A list with a single element containing a vector indicating whether hypotheses are rejected (1) or not (0).

Author(s)

Kaiyuan Hua, Xiaodong Luo

Examples

```
graphical(p = c(0.02, 0.03, 0.01))
```

---

hochbergd *Hochberg procedure*

Description

hochbergd() computes the Hochberg procedure with different alphas for different endpoints.

Usage

```
hochbergd(pvalues, alpha, epsilon = 1e-10, precision = 10)
```

Arguments

pvalues	Numeric vector of p-values from different endpoints.
alpha	Numeric vector of alpha values for the different endpoints. Vector must be same length as pvalues.
epsilon	Numeric scalar indicating the lower bound for alpha.
precision	Integer scalar of the desired number of digits to be used.

Value

List with element named decisions containing an index of rejected hypotheses.

Author(s)

Xiaodong Luo

Examples

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

`hommeld` *Hommel procedure*

Description

`hommeld()` implement the Hommel procedure with different alphas for different endpoints.

Usage

```
hommeld(pvalues, alpha, epsilon = 1e-10, precision = 10)
```

Arguments

`pvalues`	Numeric vector of p-values from different endpoints.
`alpha`	Numeric vector of alpha values for the different endpoints. Vector must be same length as `pvalues`.
`epsilon`	Numeric scalar indicating the lower bound for alpha.
`precision`	Integer scalar of the desired number of digits to be used.

Details

The package hommel can handle Hommel procedure with different alpha's for different endpoints, the function `hommeld()` is just a wrapper of `hommel::hommel()`.

Value

List with element named `decisions` containing an index of rejected hypotheses.

Author(s)

Xiaodong Luo

Examples

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

inftocor *Transform information fractions into correlation matrix*

Description

`inftocor()` transforms information (fractions) into correlation matrix.

Usage

```
inftocor(ir = c(0.2, 0.5, 1))
```

Arguments

`ir`	Numeric vector of the sequence of information fractions. All elements should be between 0 and 1 with the last one being exactly 1.

Value

List with an element named `cor` for the correlation matrix.

Author(s)

Xiaodong Luo

Examples

```
inftocor(ir = c(0.2, 0.5, 1.0))
```

seqgspgx *Sequential graphical procedure based on group-sequential p-values*

Description

`seqgspgx()` implements the sequential graphical procedure for multiple hypotheses based on group-sequential p-values.

Usage

```
seqgspgx(
  pm = matrix(rep(c(0.03, 0.04, 0.01), times = 2), ncol = 3, nrow = 2),
  alpha = 0.025,
  W = c(0.6, 0.4),
  G = rbind(c(0, 1), c(1, 0))
)
```

Arguments

pm	Numeric matrix of group-sequential p-values for different hypotheses (in row) at different times (in column).
alpha	Numeric scalar of the overall family-wise error rate.
W	Numeric vector of the weights of the graph.
G	Numeric transition matrix of the graph.

Value

List with elements

- 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

```
seqgspgx(
  pm = matrix(rep(c(0.03, 0.04, 0.01), times = 2), ncol = 3, nrow = 2),
  alpha = 0.025,
  W = c(0.6, 0.4),
  G = rbind(c(0, 1), c(1, 0))
)
```

---

seqgsphh *Sequential generalized Hochberg and Hommel procedures based on group-sequential p-values*

Description

seqgsphh() implements the sequential Generalized Hochberg and Hommel procedures based on group-sequential p-values.

Usage

```
seqgsphh(
  pm = matrix(rep(c(0.03, 0.04, 0.01), times = 2), ncol = 3, nrow = 2),
  alpha = 0.025,
  epsilon = 1e-10,
  precision = 10,
  method = "Hochberg"
)
```

Arguments

pm	Numeric matrix of group-sequential p-values for different hypotheses (in row) at different times (in column).
alpha	Numeric scalar of the overall family-wise error rate.
epsilon	Numeric scalar indicating the lower bound for alpha.
precision	Integer scalar for precision of the values, obsolete for backward compatibility.
method	"Hochberg" or "Hommel"

Value

List with elements

- 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

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

---

seqmbgx *Maurer-Bretz sequential graphical approach*

Description

seqmbgx() conducts group-sequential testing for multiple hypotheses based on Maurer-Bretz approach.

Usage

```
seqmbgx(
  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,
  sided = -1,
  W = c(0.5, 0.5, 0, 0),
  G = rbind(c(0, 0, 1, 0), c(0, 0, 0, 1), c(0, 1, 0, 0), c(1, 0, 0, 0)),
```

```
  tol = 1e-10,
  retrospective = 0
)
```

Arguments

xm	Numeric matrix of test statistics for each endpoint (in row) and each time point (in column).
informationm	Numeric matrix of information fractions for the statistics xm.
spending	Character vector for the type(s) of the spending function for each endpoint.
param.spending	parameter in the spending function
alpha	overall family-wise error rate
sided	Integer scalar indicating the side of the test: • -1: Reject if test statistic is smaller than or equal to the critical value (one-sided) • 1: Reject if test statistic is greater or equal to the critical value (one-sided) • 0: Reject if the absolute value of the test statistic is greater than the critical value (two-sided)
W	Numeric vector of the weights of the graph.
G	Numeric transition matrix of the graph.
tol	Numeric scalar of tolerance level for computing the critical values.
retrospective	Integer scalar with the following potential values • 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 retrospectively looking at the values is statistically valid in terms of control the type-1 error, not retrospectively looking at the past comparisons avoids the dilemma of retrospectively increasing the alpha level for the un-rejected hypothesis in the past.

Value

List with elements

- 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

```
seqmbgx(
  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,
  W = c(0.5, 0.5, 0, 0),
  G = rbind(c(0, 0, 1, 0), c(0, 0, 0, 1), c(0, 1, 0, 0), c(1, 0, 0, 0)),
  retrospective = 0
)
```

---

seqqvalhh *Sequential generalized Hochberg and Hommel procedures based on q-values*

Description

Sequential generalized Hochberg and Hommel procedures based on q-values

Usage

```
seqqvalhh(
  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 = 1e-10,
  precision = 10,
  method = "Hochberg"
)
```

Arguments

pm	Matrix of group-sequential p-values for different hypotheses (in row) at different times (in column).
alpham	Matrix of alpha spending corresponding to the p-values pm. For each row, alpha levels must be non-decreasing.
epsilon	Numeric scalar indicating the lower bound for alpha.
precision	Integer scalar for precision of the values, obsolete for backward compatibility.
method	Character scalar "Hochberg" or "Hommel".

Value

List with elements

- 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).

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)
seqqvalhh(pm = pm, alpham = alpham, method = "Hochberg")
seqqvalhh(pm = pm, alpham = alpham, method = "Hommel")
```

---

spendingfun *Calculate alpha spending function*

Description

spendingfun() calculates the alpha spending function.

Usage

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

Arguments

alpha	Numeric scalar of the overall alpha to be spent.
fractions	Numeric vector of the sequence of information fractions. All elements should be between 0 and 1 with the last one being exactly 1.
family	Character scalar for the family of spending functions, one of "OBF", "pocock", "power".
rho	Numeric scalar of auxiliary parameter for O'Brien-Fleming and power family.

Details

- "OBF": O'Brien-Fleming family; $2\{1 - \Phi(\Phi^{-1}(1 - \alpha/2)/t^{\rho/2})\}$;
- "pocock": Pocock family; $\alpha \log\{1 + (e - 1) * t\}$;
- "power": Power family; $\alpha * t^{\rho}$

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

Value

List with an element named aseq for the alpha spending sequence.

Author(s)

Xiaodong Luo

Examples

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

---

updategraph *Update graph*

Description

`updategraph()` updates the graph when only a subset of original hypotheses is concerned.

Usage

```
updategraph(
  S1 = c(2, 3),
  W0 = c(0.5, 0.5, 0, 0),
  G0 = rbind(c(0, 0, 1, 0), c(0, 0, 0, 1), c(0, 1, 0, 0), c(1, 0, 0, 0)),
  S0 = seq(1, length(W0), by = 1)
)
```

Arguments

`S1`	Integer indices of the subset of hypotheses, S1 must be a non-empty subset of `S0` and must be sorted increasingly.
`W0`	Numeric vector for the initial weights of the graph.
`G0`	Numeric matrix of dimesion `length(W0)` by `length(W0)` for the initial transition matrix of the graph.
`S0`	Integer indices for the set of hypotheses from 1 to length of `W0`.

Value

List with the following elements

- `S1`: Integer indices the same as the input `S1`.
- `W1`: Numeric vector for weights of the updated graph.
- `G1`: Numeric transition of the updated graph.

Author(s)

Xiaodong Luo

Examples

```
## We can use the function to produce a closed testing tree
## A function to create power set
powerset <- function(x) {
  sets <- lapply(1:(length(x)), function(i) combn(x, i, simplify = FALSE))
  unlist(sets, recursive = FALSE)
}

n <- 3    # number of hypotheses
pn <- 2^n-1
pset <- powerset(seq(1, n, by = 1))    # create the power set
df <- data.frame(matrix(ncol = 1+n, nrow = 0))    # create the dataset
colnames(df) <- c("Test", paste0("H", seq(1, n, by = 1), sep = ""))

W0 <- c(1/3, 1/3, 1/3)    # the weights of the graph
m <- rbind(H1 = c(0, 1/2, 1/2),
           H2 = c(1/2, 0, 1/2),
           H3 = c(1/2, 1/2, 0))
G0 <- matrix(m, nrow = 3, ncol = 3)    # the transition matrix of the graph

for (j in 1:pn){
    abc <- updategraph(S1 = pset[[j]], W0 = W0, G0 = G0)
    temp <- rep("-", n)
    temp[pset[[j]]] <- abc$W1
    temp <- c(paste(pset[[j]], collapse = ""), temp)
    df[j, ] <- temp
}
df    # the dataframe lists the closed testing tree
```

Index