Package 'someKfwer'

February 20, 2015

Type Package

Title Controlling the Generalized Familywise Error Rate

Version 1.2

Date 2009-10-30

Author L. Finos and A. Farcomeni

Maintainer L. Finos <livio@stat.unipd.it>

Description This package collects some procedures controlling the Generalized Familywise Error Rate.

License GPL (>= 2)

LazyLoad yes

Repository CRAN

Date/Publication 2014-11-24 15:40:43

NeedsCompilation no

R topics documented:

---

someKFWER-package   *Controlling the k-FWER (Generalized Familywise Error Rate)*

Description

This package collects some procedures controlling the Generalized Familywise Error Rate: Lehmann and Romano (2005), Guo and Romano (2007) (single step and stepdown), Finos and Farcomeni (2009).

Details

Package:	kfwe
Type:	Package
Version:	1.0
Date:	2009-10-30
License:	GPL (>= 2)
LazyLoad:	yes

Author(s)

L. Finos and A. Farcomeni

Maintainer: <livio@stat.unipd.it>

References

Finos and Farcomeni (2010) k-FWER control without multiplicity correction, with application to detection of genetic determinants of multiple sclerosis in Italian twins. Biometrics (Articles online in advance of print: DOI 10.1111/j.1541-0420.2010.01443.x)

Examples

```
set.seed(13)
y=matrix(rnorm(3000),3,1000)+2                    #create toy data
p=apply(y,2,function(y) t.test(y)$p.value)        #compute p-values
M2=apply(y^2,2,mean)                              #compute ordering criterion

kord=kfweOrd(p,k=5,ord=M2)                        #ordinal procedure
kgr=kfweGR(p,k=5)                                 #Guo and Romano

kord=kfweOrd(p,k=5,ord=M2,GD=TRUE)                #ordinal procedure (any dependence)
klr=kfweLR(p,k=5)                                #Lehaman and Romano (any dependence)
```

`Procedures controlling the k-FWER (Generalized Familywise Error Rate)`
Controlling the Generalized Familywise Error Rate

Description

This library collects some procedures controlling the Generalized Familywise Error Rate: Lehmannn and Romano (2005), Guo and Romano (2007) (single step and stepdown), Finos and Farcomeni (2009).

Usage

```
kfweLR(p, k = 1, alpha = 0.01, disp = TRUE)
kfweGR(p, k = 1, alpha = 0.01, disp = TRUE,
SD=TRUE, const = 10, alpha.prime = getAlpha(k = k, s = length(p),
                    alpha = alpha, const = const))
kfweOrd(p, k = 1, alpha = 0.01, ord = NULL,
    alpha.prime = alpha, J = qnbinom(alpha,k,alpha.prime),
    disp = TRUE, GD=FALSE)

getAlpha (s, k = 1, alpha = 0.01, const = 10)
```

Arguments

`p`	vector of p-values of length s
`s`	number of p-values (i.e. hypotheses)
`k`	number of allowed errors in kFWE controls
`alpha`	global significance level
`ord`	the vector of values based on which the p-values have to be ordered
`const`	Bigger is better (more precise but slower)
`J`	number of allowed jumps befor stopping
`disp`	diplay output? TRUE/FALSE
`SD`	Step-down version of the procedure? (TRUE/FALSE) the step-down version is uniformly more powerful than the single step one.
`alpha.prime`	univariate alpha for single step Guo and Romano procedure
`GD`	Logic value. Should the correction for general dependence be applied? (See reference below for further details)

Value

kfweOrd, kfweLR, kfweGR, kfweGR.SD return a vector of kFWE-adjusted p-values. It respect the order of input vector of p-values p.

getAlpha returns the alpha for Guo and Romano procedure.

Author(s)

L. Finos and A. Farcomeni

References

For Lehmann and Romano procedure see:

Lehmann and Romano (2005) Generalizations of the Familywise Error Rate, Annals of Statistics, 33, 1138-1154.

For Guo and Romano procedure see:

Guo and Romano (2007) A Generalized Sidak-Holm procedure and control of genralized error rates under independence, Statistical Applications in Genetics and Molecular Biology, 6, 3.

For Ordinal procedure see:

Finos and Farcomeni (2010) k-FWER control without multiplicity correction, with application to detection of genetic determinants of multiple sclerosis in Italian twins. Biometrics (Articles online in advance of print: DOI 10.1111/j.1541-0420.2010.01443.x)

Examples

```
set.seed(13)
y=matrix(rnorm(3000),3,1000)+2                    #create toy data
p=apply(y,2,function(y) t.test(y)$p.value)        #compute p-values
M2=apply(y^2,2,mean)                              #compute ordering criterion

kord=kfweOrd(p,k=5,ord=M2)                        #ordinal procedure
kgr=kfweGR(p,k=5)                                 #Guo and Romano

kord=kfweOrd(p,k=5,ord=M2,GD=TRUE)                #ordinal procedure (any dependence)
klr=kfweLR(p,k=5)                                #Lehaman and Romano (any dependence)
```

Index