Package 'someMTP'

October 14, 2022

Type Package
Title Some Multiple Testing Procedures
Version 1.4.1.1
Date 2013-11-04
Author livio finos
Maintainer livio finos <livio@stat.unipd.it>
Depends methods
Description It's a collection of functions for Multiplicity Correction and Multiple Testing.
License GPL (>= 2)
LazyLoad yes
NeedsCompilation no
Repository CRAN
Date/Publication 2021-03-01 07:10:10 UTC

R topics documented:

someMTP-package ........................................ 2
*OrNULL-class ........................................... 3
draw ...................................................... 3
fdrOrd/kfweOrd ......................................... 4
lsd.object class ......................................... 6
lsd.test .................................................. 7
p.adjust.w ............................................... 8
someMTP.object class ................................... 10
step.adj .................................................. 11

Index 13
someMTP-package  Some Multiple Testing Procedures

Description

It is a collection of functions for Multiplicity Correction and Multiple Testing.

Details

Package: someMTP
Type: Package
Version: 1.2
Date: 2011-01-10
License: GPL (>= 2)
LazyLoad: yes

Author(s)

livio finos
Maintainer: <livio@stat.unipd.it>

References

For weighted methods:

For LSD test:

Examples

```r
set.seed(13)
y <- matrix(rnorm(5000),5,1000) #create toy data
y[,1:100] <- y[,1:100]+3 #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
```
\texttt{fdr} <- \texttt{p.adjust(p, method="BH")} \# (unweighted) procedure, fdr control 
\texttt{sum(fdr<.05)}
\texttt{fdr.w <- p.adjust.w(p, method="BH", w=M2)} \# weighted procedure, weighted fdr control 
\texttt{sum(fdr.w<.05)}
\texttt{fwer} <- \texttt{p.adjust(p, method="holm")} \# (unweighted) procedure, fwer control 
\texttt{sum(fwer<.05)}
\texttt{fwer.w <- p.adjust.w(p, method="BHfwe", w=M2)} \# weighted procedure, weighted fwer (=fwer) control 
\texttt{sum(fwer.w<.05)}
\texttt{plot(M2,-log10(p))}

\*OrNULL-class

\textbf{Description}

\begin{itemize}
  \item class * or Null
\end{itemize}

\textbf{Objects from the Class}

A virtual Class: No objects may be created from it.

\textbf{Methods}

No methods defined with class "*OrNULL" in the signature.

\textbf{Examples}

\begin{verbatim}
showClass("callOrNULL")
\end{verbatim}

\textbf{draw}

\textit{Plots results of fdrOrd()}

\textbf{Description}

Plots results of fdrOrd()

\textbf{Usage}

\begin{verbatim}
draw(object, what = c("all", "ordVsP", "stepVsR"), pdfName = NULL)
\end{verbatim}

\textbf{Arguments}

\begin{itemize}
  \item \texttt{object} \hspace{2cm} a \texttt{someMTP.object} resulting from fdrOrd()
  \item \texttt{what} \hspace{2cm} what to plot; "all" is the default
  \item \texttt{pdfName} \hspace{2cm} it is the pdf filename where the plot will be saved. If pdfName is null (the default) the plot will show as window.
\end{itemize}
**fdrOrd/kfweOrd**

*Controlling the False Discovery Rate and the Generalized FWER in ordered Test*

**Value**

No value is returned

**Author(s)**

Livio Finos

**See Also**

See Also `fdrOrd`.

**Examples**

```r
set.seed(17)
x = matrix(rnorm(60), 3, 20)
x[, 1:10] = x[, 1:10] + 2  # variables 1:10 have tests under H1
ts = apply(x, 2, function(x) t.test(x)$statistic)
p = apply(x, 2, function(x) t.test(x)$p.value)
m2 = apply(x^2, 2, mean)
pOrd <- fdrOrd(ps, q = .05, ord = m2)
draw(pOrd)
```

---

**Description**

Ordinal procedure controlling the FDR and the Generalized FWER

**Usage**

```r
fdrOrd(p, q = .01, ord = NULL, GD = FALSE)
kfweOrd(p, k = 1, alpha = 0.01, ord = NULL, alpha.prime = alpha,
        J = qnbinom(alpha, k, alpha.prime), GD = FALSE)
```

**Arguments**

- `p`: vector of p-values
- `ord`: Values on the basis of which the procedure select the hypotheses (following decreasing order). The vector have the same length of `p`. If NULL the natural ordering is considered.
- `q`: average FDR level
- `alpha`: global significance level
- `k`: number of allowed errors in kFWE controls
- `J`: number of allowed jumps before stopping
alpha.prime  univariate alpha for single step Guo and Romano procedure
GD          Logic value. Should the correction for general dependence be applied?

**Value**

The function returns an object of class `someMTP.object`.

- `rej`: a logical vector indicating whenever the related hypotheses have been rejected.
- `p`: the vector of p-values used in the call
- `ord`: The vector used to sort the p-values (decreasing).
- `MTP`: "fdrOrd" or "kfweOrd"
- `GD`: A logical value indicating if the correction for General Dependence have been used or not.
- `q`: The level of controlled FDR.
- `alpha`: The level of controlled k-FWER
- `alphaprime`: The significance level of individual tests
- `k`: Number of allowed Errors
- `J`: Number of allowed Jumps

**Author(s)**

L. Finos and A. Farcomeni

**References**


**See Also**

See also `draw`

**Examples**

```r
set.seed(17)
x = matrix(rnorm(60), 3, 20)
x[, 1:10] = x[, 1:10] + 2  # variables 1:10 have tests under H1
ts = apply(x, 2, function(x) t.test(x)$statistic)
ps = apply(x, 2, function(x) t.test(x)$p.value)  # compute p-values
m2 = apply(x[, 2], 2, mean)  # compute ordering criterion
pOrd <- fdrOrd(ps, q = .05, ord = m2)  # ordinal Procedure
pOrd
```
draw(pOrd)
sum(p.adjust(ps, method="BH")<=.05) # rejections with BH

kOrd <- kfweOrd(ps, k=5, ord=m2) # ordinal procedure
kOrd
kOrdGD <- kfweOrd(ps, k=5, ord=m2, GD=TRUE) # ord. proc. (any dependence)
kOrdGD

## lsd.object class

Class "lsd.object" for storing the result of the function lsd

**Description**

The class lsd.object is the output of a call to `lsd.test`

**Slots**

- `F`: the test statistic
- `df`: the degrees of freedom of F
- `globalP`: the associated p-value
- `D`: the matrix used in the test (it provides the influence of columns in resp to the test statistic)
- `call`: The matched call to `lsd`
- `MTP`: The procedure used ("fdrOrd", "kfweOrd" or others).

**Methods**

- `p.value` (`lsd.object`): Extracts the p-values.
- `show` `lsd.object`: Prints the test results: p-value, test statistic, expected value of the test statistic under the null hypothesis, standard deviation of the test statistic under the null hypothesis, and number of covariates tested.
- `summary` `lsd.object`: Prints the test results: p-value, test statistic, expected value of the test statistic under the null hypothesis, standard deviation of the test statistic under the null hypothesis, and number of covariates tested.
- `weights` `lsd.object`: diagonal of matrix D used in the test (i.e. the influence of columns in resp to the test statistic)

**Author(s)**

Livio Finos: `<livio@stat.unipd.it>`

**See Also**

- `lsd`
Examples

```r
# Simple examples with random data here
set.seed(1)
# Standard multivariate LSD test for one sample case
X = matrix(rnorm(50), 5, 10) + 5
res <- lsd.test(resp = X, alternative = ~ 1)
print(res)
p.value(res)
summary(res, showD = TRUE)
```

---

**lsd.test**

*Multivariate Left Spherically Distributed (LSD) linear scores test.*

**Description**

It performs the multivariate Left Spherically Distributed linear scores test of L"auter et al. (The Annals of Statistics, 1998) (see also details below).

**Usage**

```r
lsd.test(resp, alternative = 1, null = NULL, D = NULL, data = NULL)
```

**Arguments**

- `resp` The response vector of the regression model. May be supplied as a vector or as a formula object. In the latter case, the right hand side of `Y` is passed on to `alternative` if that argument is missing, or otherwise to `null`.
- `alternative` The part of the design matrix corresponding to the alternative hypothesis. The covariates of the null model do not have to be supplied again here. May be given as a half formula object (e.g. `~ a + b`). In that case the intercept is always suppressed.
- `null` The part of the design matrix corresponding to the null hypothesis. May be given as a design matrix or as a half formula object (e.g. `~ a + b`). The default for `Z` is `~ 1`, i.e. only an intercept. This intercept may be suppressed, if desired, with `Z = ~ 0`.
- `data` Only used when `Y`, `X`, or `Z` is given in formula form. An optional data frame, list or environment containing the variables used in the formulae. If the variables in a formula are not found in `data`, the variables are taken from `environment(formula)`, typically the environment from which `gt` is called.
- `D` is a q x p matrix or it is a function with arguments `resp` and `null` returning the q x p transformation matrix. When `D = NULL`, then `D = diag(t(resp))%*%IP0%*%resp)` with `IP0 = diag(n) - null%*%solve(t(null)%*%null)%*%t(null)`
The function returns an object of class lsd.object.

<table>
<thead>
<tr>
<th>F</th>
<th>the test statistic</th>
</tr>
</thead>
<tbody>
<tr>
<td>df</td>
<td>the degrees of freedom of F</td>
</tr>
<tr>
<td>p</td>
<td>the associated p-value</td>
</tr>
<tr>
<td>D</td>
<td>the matrix used in the test (it provide information on the influence of columns in resp to the test)</td>
</tr>
</tbody>
</table>

call: The matched call to lsd.test.

Author(s)

Livio Finos

References


Examples

```r
set.seed(1)
#Standard multivariate LSD test for one sample case
X=matrix(rnorm(50),5,10)+2
lsd.test(resp=X,alternative=-1)

#Standard multivariate LSD test for two sample case
X2=X+matrix(c(0,0,1,1,1),5,10)*10
lsd.test(resp=X2,null=~1,alternative=c(0,1,1,1,1))

#General multivariate LSD test for linear predictor with covariates
lsd.test(resp=X2,null=cbind(rep(1,5),c(0,0,1,1,1)),alternative=1:5)
```

---

p.adjust.w

Adjust P-values for Multiple Comparisons

Description

Given a set of p-values, returns p-values adjusted using one of several (weighted) methods. It extends the method of p.adjust(stats)

Usage

```r
p.adjust.w(p, method = c("bonferroni","holm","BHfwe","BH","BY"), n = length(p),w=NULL)
```
Arguments

- p: vector of p-values (possibly with NAs)
- method: correction method
- n: number of comparisons, must be at least length(p); only set this (to non-default) when you know what you are doing!
- w: weights to be used. p.adjust.w(..., rep(1, length(p))) produces the same results as in p.adjust(...) (i.e. the unweighted counterpart).

Value

A vector of corrected p-values (same length as p) having two attributes: attributes(...)$w is the vector of used weights and attributes(...)$method is the method used.

Author(s)

Livio Finos

References


See Also

p.adjust

Examples

```R
set.seed(13)
y <- matrix(rnorm(5000),5,1000) #create toy data
y[,1:100] <- y[,1:100]+3 #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
fdr <- p.adjust(p,method="BH") #(unweighted) procedure, fdr control
sum(fdr<.05)
fdr.w <- p.adjust.w(p,method="BH",w=M2) #weighted procedure, weighted fdr control
sum(fdr.w<.05)

fwer <- p.adjust(p,method="holm") #(unweighted) procedure, fwer control
sum(fwer<.05)
fwer.w <- p.adjust.w(p,method="BHfwe",w=M2) #weighted procedure, weighted fwer (=fwer) control
sum(fwer.w<.05)

plot(M2,-log10(p))
```
Class "someMTP.object" for storing the result of the function fdrOrd

Description

The class someMTP.object is the output of a call to fdrOrd. It also stores the information needed for related plots.

Slots

rej: a logical vector indicating whenever the related hypotesis have been rejected.
p: The vector of (raw) p-values used in the procedure.
ord: The vector used to sort the p-values (decreasing).
idOrd: The vector of indices used in sorting.
MTP: The type of procedure used.
GD: A logical value incating if the correction for General Dependence have been used or not.
q: The level of contrelled FDR when MTP=="fdrOrd".
k: The number of false rejection when MTP=="kfweOrd"
J: The number of allowed Jumps when MTP=="kfweOrd"
alpha: The significance level when MTP=="kfweOrd"
alphaprime: The significance level of individual tests.
call: The cal that generates the object.

Methods

show someMTP.object: Prints the test results.
summary someMTP.object: Prints the test results (as show).
draw someMTP.object: Plots results; what = c("all", "ordVsP", "stepVsR")
sort signature(x = "someMTP.object"): Sorts the p-values to decreasing order of ord.
length signature(x = "someMTP.object"): The number of tests performed.
names signature(x = "someMTP.object"): Extracts the row names of the results matrix.
names<- signature(x = "someMTP.object"): Changes the row names of the results matrix. Du-
plicate names are not allowed, but see alias.

Author(s)

Livio Finos: <livio@stat.unipd.it>

See Also

someMTP.object
**Examples**

```r
# Simple examples with random data
set.seed(17)
x <- matrix(rnorm(60), 3, 20)
x[, 1:10] = x[, 1:10] + 2  ## variables 1:10 have tests under H1
ts = apply(x, 2, function(x) t.test(x)$statistic)
p = apply(x, 2, function(x) t.test(x)$p.value)
m2 = apply(x^2, 2, mean)
pOrd <- fdrOrd(p, q = .05, ord = m2)
pOrd
length(pOrd)
names(pOrd) <- paste("V", 1:20, sep="")
names(pOrd)
```

**Description**

Corrects the p-value due to model selection. It works with models of class `glm` and selected with function `step` (stats).

**Usage**

```r
step.adj(object, MC = 1000, scope = NULL, scale = 0,
direction = c("both", "backward", "forward"),
trace = 0, keep = NULL, steps = 1000, k = 2)
```

**Arguments**

- `object` object of class `glm`. Note that formula have to write by variables name like `y~var1+var2+var3`. `data` is a data.frame (see example below). `offset` is not yet implemented, avoid its use, `glm(formula, data, family=gaussian)` produce the same result of `lm(formula, data)`, then linear model can be allways performed
- `MC` number of random permutations for the dependent variable
- `scope` as in function `step`
- `scale` as in function `step`
- `direction` as in function `step`
- `trace` as in function `step`
- `keep` as in function `step`
- `steps` as in function `step`
- `k` as in function `step`, other arguments are not implemented yet.
Details

It performs anova function (stats library) on the model selected by function step vs the null model with the only intercept and it corrects for multiplicity. For lm models and gaussian glm models it computes a F-test, form other models it uses Chisquare-test (see also anova.glm and anova.lm help).

Value

An anova table with an extra column reporting the corrected p-value

Author(s)

Livio Finos and Chiara Brombin

References


See Also

glm, anova

Examples

set.seed(17)
y=rnorm(10)
x=matrix(rnorm(50),10,5)
#define a data.frame to be used in the glm function
DATA=data.frame(y,x)
#fit the model on a toy dataset
mod=glm(y~X1+X2+X3+X4+X5,data=DATA)

#select the model using function step
mod.step=step(mod, trace=0)
#test the selected model vs the null model
anova(glm(y~1, data=DATA),mod.step,test="F")

#step.adj do the same, but it also provides multiplicity control
step.adj(mod,MC=101, trace=0)
Index

* classes
  *OrNULL-class, 3

* htest
  fdrOrd/kfweOrd, 4
  lsd.test, 7
  p.adjust.w, 8
  step.adj, 11

* methods
  lsd.object class, 6
  someMTP.object class, 10

* package
  someMTP-package, 2
  *OrNULL-class, 3

anova, 12

callOrNULL-class (*OrNULL-class), 3
characterOrNULL-class (*OrNULL-class), 3

draw, 3, 5

fdrOrd, 4, 10
fdrOrd/fdrOrd/kfweOrd, 4
fdrOrd/kfweOrd, 4
formula, 7

glm, 12

kfweOrd (fdrOrd/kfweOrd), 4

length (someMTP.object class), 10
length, someMTP.object-method (someMTP.object class), 10
length-method (someMTP.object class), 10
listOrNULL-class (*OrNULL-class), 3
lsd, 6

lsd (lsd.test), 7
lsd.object (lsd.object class), 6
lsd.object class, 6
lsd.object-class (lsd.object class), 6

lsd.test, 6, 7, 8

matrixOrNULL-class (*OrNULL-class), 3

names, someMTP.object-method (someMTP.object class), 10
names<-, someMTP.object-method (someMTP.object class), 10
numericOrNULL-class (*OrNULL-class), 3

p.adjust, 9
p.adjust.w, 8
p.value (lsd.object class), 6
p.value, lsd.object-method (lsd.object class), 6

show, lsd.object-method (lsd.object class), 6
show, someMTP.object-method (someMTP.object class), 10
someMTP (someMTP-package), 2
someMTP-package, 2
someMTP.object (someMTP.object class), 10
someMTP.object class, 10
someMTP.object-class (someMTP.object class), 10

sort, someMTP.object-method (someMTP.object class), 10

step.adj, 11

summary (lsd.object class), 6
summary, lsd.object-method (lsd.object class), 6

summary, someMTP.object-method (someMTP.object class), 10
summary-method (someMTP.object class), 10

vectorOrNULL-class (*OrNULL-class), 3

weights (lsd.object class), 6
weights, lsd.object-method (lsd.object class), 6