Package 'multfisher'

October 13, 2022

Title Optimal Exact Tests for Multiple Binary Endpoints

Version 1.1

Description Calculates exact hypothesis tests to compare a treatment and a reference group with respect to multiple binary endpoints.
The tested null hypothesis is an identical multidimensional distribution of successes and failures in both groups. The alternative
hypothesis is a larger success proportion in the treatment group in at least one endpoint. The tests are based on the multivariate
permutation distribution of subjects between the two groups. For this permutation distribution, rejection regions are calculated
that satisfy one of different possible optimization criteria. In particular, regions with maximal exhaustion of the nominal
significance level, maximal power under a specified alternative or maximal number of elements can be found. Optimization is achieved
by a branch-and-bound algorithm. By application of the closed testing principle, the global hypothesis tests are extended to multiple
testing procedures.

Date 2018-02-23

Author Robin Ristl [aut, cre]

Maintainer Robin Ristl <robin.ristl@meduniwien.ac.at>

Depends R (>= 3.0.0)

Imports stats

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2018-02-23 10:29:06 UTC

R topics documented:

mfisher.test 2
plot.multfisher 5
print.multfisher 7

Index **8**

`mfisher.test` *Optimal Exact Tests for Multiple Binary Endpoints*

Description

Calculates global tests and multiple testing procedures to compare two groups with respect to multiple binary endpoints based on optimal rejection regions.

Usage

```
mfisher.test(x, y = NULL, method = c("alpha.greedy", "alpha", "number",
  "power", "bonferroni.greedy"), alpha = 0.025, p1 = NULL, p0 = NULL,
  max.iter = 10^5, limit = 0, show.region = FALSE, closed.test = FALSE,
  consonant = FALSE)
```

Arguments

`x`	a data frame of binary response vectors, or an array of numbers of failures and successes in the treatment group, or a list of marginal *2 by 2* tables, see details.
`y`	a vector of group allocations, or an array of numbers of failures and successes in the reference group, see details.
`method`	a character variable indicating which optimization procedure to use. This can be one of `"alpha.greedy"`, `"alpha"`, `"number"`, `"power"` or `"bonferroni.greedy"`, see details.
`alpha`	nominal significance level, the default is 0.025. Note that the test is one-sided.
`p1`	an array of assumed probabilities for failure and success in the treatment group, see details.
`p0`	an array of assumed probabilities for failure and success in the reference group, see details.
`max.iter`	the maximal number of iterations in the branch and bound optimzation algorithm. Defaults to 10^5.
`limit`	the value below which contributions to alpha are set to zero (and alpha is lowered accordingly) to speed up computation. Defaults to 0.
`show.region`	logical, if `TRUE` a data frame indicating which possible outcome is element of the rejection region of the global test is added to the output. Defaults to `FALSE`.

`closed.test` logical, if `TRUE` adjusted p-values for the elementary null hypotheses are calculated by applying the specified test to all intersection hypotheses in a closed testing scheme. This can be computer intensive, depending on the number of endpoints.

`consonant` logical indicating if the global test should be constrained such that the resulting closed test is consonant. This option is only available for two endpoints. Note that the Bonferroni greedy method is always consonant by construction.

Details

The null hypothesis for the global test is an identical multidimensional distribution of successes and failures in both groups. The alternative hypothesis is a larger success proportion in the treatment group in at least one endpoint.

`x` can be a data frame with one row per subject and one column for each endpoint. Only values of 0 or 1 are allowed, with 0 indicating failure and 1 indicating success of the subject for the particular endpoint. In that case `y` needs to be a vector of group assignemnts with values 0 and 1, where 0 is the reference group and 1 the treatment group. Alternatively, `x` and `y` can be contingency tables in terms of *2 by 2 by ... by 2* arrays. Each dimension of the array corresponds to one endpoint, the first coordinate position in each dimension refers to failure in that endpoint, the second coordinate position refers to success. The array contains the number of subjects that were observed for each possible combination of failures and successes. If `x` is a list of marginal *2 by 2* tables, the Bonferroni greedy method is used. Matching the other input variants, the *2 by 2* tables are assumed to have the number of failures in the first row and the number of successes in the second row, and the first column to correspond to the reference group, the second column to the treatment group.

The methods `"alpha.greedy"`, `"alpha"`, `"number"` and `"power"` are based on the multivariate permutation distribution of the data conditional on the observed numbers of successes and failures across both groups. The method `"alpha.greedy"` uses a greedy algorithm aiming to exhaust the nominal significance level. The methods `"alpha"`, `"number"` and `"power"` use a branch and bound algorithm to find rejection regions with, respectively, maximal exhaustion of the nominal significance level, maximal number of elements or maximal power for the alternative given by `p1` and `p0`. The method `"bonferroni.greedy"` uses a greedy algorithm aiming to exhaust the nominal significance level of a weighted Bonferroni adjustment of multiple Fisher's exact tests. See reference for further details.

`p1` and `p0` are *2 by 2 by ... by 2* arrays. Each dimension of the array corresponds to one endpoint, the first coordinate position in each dimension refers to failure in that endpoint, the second coordinate position refers to success. The array contains the assumed true probabilities for each possible combination of failures and successes.

Value

A list with class `multfisher` containing the following components:

`call` the function call.

`data` a data frame showing the aggregated input data. If `p1` and `p0` are provided they are included in vectorized form.

`alpha` the value of `alpha`.

`method` the chosen method as found by argument match to `method`.

`statistic` the vector of test statistics, these are the marginal numbers of successes in the treatment group.

`p.value` the p-value of the global test. See reference for details on the calculation.

`conditional.properties` a list of the actual significance level, the number of elements and the power of the global test. The values are calculated from the permutation distribution of the date and they are conditional on the observed total numbers of successes and failures. The power is calculated for the alternative defined through `p1` and `p0`. If `p1` and `p0` are not specified, the value for power is `NA`.

`rej.region` Provided if `show.region` is `TRUE` and method is in `c("alpha","number","power","alpha.greedy")`. A data frame showing in the column rejection.region if a multidimensional test statistic, indicated by the previous columns, is element of the rejection region (value of 1) or not (value of 0) for the global level alpha test. The column alpha gives the probability of observing the particular vector of test statistics under the null hypothesis and conditional on the observed total numbers of successes and failures. Values of 0 occur if a combination of test statistics is not possible in the conditional distribution. The column power shows the conditional probability under the alternative defined through `p1` and `p0`. If `p1` and `p0` are not specified, the values for power are `NA`.

`elementary.tests` a data frame showing for each endpoint the marginal odds ratio, the unadjusted one-sided p-value of Fisher's exact test and the adjusted p-value resulting from application of the optimal exact test in a closed testing procedure.

`closed.test` a data frame indicating all intersection hypotheses in the closed test and giving their p-values.

`consonant.constraint` logical indicating whether the consonance constraint was used.

`OPT` a list summarizing the optimization success, if applicable. The number of iterations of the branch and bound algorithm is given, as well as the specified maximal iteration number and a logical variable indicating whether the optimization (in all steps of the closed test, if applicable) was finished. The number of iterations may be 0, which indicates that the optimization problem was solved in a pre-processing step.

Author(s)

Robin Ristl, <robin.ristl@meduniwien.ac.at>

References

Robin Ristl, Dong Xi, Ekkehard Glimm, Martin Posch (2018), Optimal exact tests for multiple binary endpoints. *Computational Statistics and Data Analysis*, **122**, 1-17. doi: 10.1016/j.csda.2018.01.001 (open access)

See Also

`print.multfisher`, `plot.multfisher`

Examples

```
## Examples with two endpoints
data<-data.frame(endpoint1=c(0,0,1,1,1,0,0,0,0,1,1,1,1,1,1, 0,0,1,0,0,1,1,1,1,1,1,1,1,1,1),
```

```
  endpoint2=c(0,0,0,0,0,1,1,1,1,1,1,1,1,1,1, 0,0,0,1,1,1,1,1,1,1,1,1,1,1,1),
  group=rep(c(0,1),each=15))
## maximal power under a specified alternative
p1<-matrix(c(0.1,0.2,0.2,0.5),2,2)
p0<-matrix(c(0.75,0.1,0.1,0.05),2,2)
rownames(p1)<-rownames(p0)<-c("EP1_failure","EP1_success")
colnames(p1)<-colnames(p0)<-c("EP2_failure","EP2_success")
testpower<-mfisher.test(x=data[,c(1:2)],y=data$group,method="power",
p1=p1,p0=p0,closed.test=TRUE,show.region=TRUE)
print(testpower)
plot(testpower,cex=2)
str(testpower)

## maximal alpha with consonance constraint and using aggregated data as input
tab1<-table(data$endpoint1[data$group==1],data$endpoint2[data$group==1])
tab0<-table(data$endpoint1[data$group==0],data$endpoint2[data$group==0])
testalpha<-mfisher.test(x=tab1,y=tab0,method="alpha",closed.test=TRUE,
show.region=TRUE,consonant=TRUE)
print(testalpha)
plot(testalpha,cex=2)

## Examples with three endpoints
data3EP<-data.frame(endpoint1=c(0,0,0,0,0,1,1,0,0,0, 0,0,0,0,1,1,1,1,1,1),
     endpoint2=c(0,0,0,0,0,1,0,1,0,0, 0,0,1,1,1,1,1,1,1,1),
     endpoint3=c(0,0,0,0,0,0,0,0,1,1, 0,0,0,1,1,1,1,1,1,1),
     group=rep(c(0,1),each=10))

## greedy alpha exhaustion
testgreedy3EP<-mfisher.test(x=data3EP[,1:3],y=data3EP$group,method="alpha.greedy",
show.region=TRUE,closed.test=TRUE)
print(testgreedy3EP)
par(mfrow=c(3,3))
for(i in 1:9) {
plot(testgreedy3EP,dim=c(1,2),slice=list(T3=i),show.titles=FALSE,cex=2,xlim=c(0,8),ylim=c(0,10))
title(paste("T3 =",i))
}

## Bonferroni greedy
mfisher.test(x=data3EP[,1:3],y=data3EP$group,method="bonferroni.greedy",closed.test=TRUE)
## Bonferroni greedy with alternative input of marginal tables
mfisher.test(x=list(table(data3EP$endpoint1,data3EP$group),
table(data3EP$endpoint2,data3EP$group),table(data3EP$endpoint3,data3EP$group)),
method="bonferroni.greedy",closed.test=TRUE)
```

`plot.multfisher` *Plot Rejection Region from a* `multfisher` *Object*

Description

Plot two dimensions of the rejection region.

Usage

```
## S3 method for class 'multfisher'
plot(x, dim = c(1, 2), slice = NULL,
  show.titles = TRUE, ...)
```

Arguments

x	an object of class multfisher
dim	a vector of length two, indicating which two dimensions of the rejection region to plot. The default is c(1,2). The argument is ignored if x was calculated for only one endpoint.
slice	a named list of numeric values at which the test statistics for the dimensions (assumedly) not included in dim are held constant, see details.
show.titles	logical indicating whether the plot title and explanatory subtitles are shown.
...	further arguments passed to the generic plot function.

Details

The function produces plots of the multivariate rejection regions calculated by mfisher.test. For more than two dimensions, the default slice=NULL shows a projection of the rejection region on the two dimensions indicated in dim. slice may be specified to produce plots of slices through the multivariate rejection region. In that case slice must be a named list of numeric objects. The names must be of the form Ti, where i is replaced by the number of the dimension to be indicated. The numeric value defines at which value the test statistic for the indicated dimension is held constant. (If there are dimensions that are neither indicated in dim nor in slice, the plot is still a projection.)

Author(s)

Robin Ristl, <robin.ristl@meduniwien.ac.at>

See Also

mfisher.test, print.multfisher

Examples

```
## Example with two endpoints
data<-data.frame(endpoint1=c(0,0,1,1,1,0,0,0,0,1,1,1,1,1,1, 0,0,1,0,0,1,1,1,1,1,1,1,1,1,1),
endpoint2=  c(0,0,0,0,0,1,1,1,1,1,1,1,1,1,1, 0,0,0,1,1,1,1,1,1,1,1,1,1,1,1),
group=rep(c(0,1),each=15))
plot(mfisher.test(x=data[,c(1:2)],y=data$group,show.region=TRUE),cex=2)

## Example with three endpoints
data3EP<-data.frame(endpoint1=  c(0,0,0,0,0,1,1,0,0,0, 0,0,0,0,1,1,1,1,1,1),
  endpoint2=  c(0,0,0,0,0,1,0,1,0,0, 0,0,1,1,1,1,1,1,1,1),
  endpoint3=  c(0,0,0,0,0,0,0,0,1,1, 0,0,0,1,1,1,1,1,1,1),
  group=rep(c(0,1),each=10))
testgreedy3EP<-mfisher.test(x=data3EP[,1:3],y=data3EP$group,method="alpha.greedy",
show.region=TRUE,closed.test=TRUE)
```

```
## Projecion on the first two dimensions
plot(testgreedy3EP,dim=c(1,2),cex=2)
## Slice at a value of 5 for the third dimension
plot(testgreedy3EP,dim=c(1,2),slice=list(T3=5),cex=2)
## Show all slices through the third dimension
par(mfrow=c(3,3))
for(i in 1:9) {
plot(testgreedy3EP,dim=c(1,2),slice=list(T3=i),show.titles=FALSE,cex=2,xlim=c(0,8),ylim=c(0,10))
title(paste("T3 =",i))
}
```

`print.multfisher` *Print Values from a* `multfisher` *Object*

Description

Print the test results.

Usage

```
## S3 method for class 'multfisher'
print(x, ...)
```

Arguments

`x`	an object of class `multfisher`
`...`	further arguments passed to other methods. Not used.

Author(s)

Robin Ristl, <robin.ristl@meduniwien.ac.at>

See Also

`mfisher.test`, `plot.multfisher`

Index

mfisher.test, 2, *6*, *7*

plot, *6*
plot.multfisher, *4*, 5, *7*
print.multfisher, *4*, *6*, 7