Package 'CrossScreening'

March 20, 2017

Type Package

Title Cross-Screening in Observational Studies that Test Many Hypotheses

Version 0.0.1

Date 2017-03-20

Description Cross-screening is a new method that uses both random halves of the sample to screen and test many hypotheses. It generally improves statistical power in observational studies when many hypotheses are tested simultaneously.

Imports stats, sensitivitymw, parallel, plyr, tables

Suggests

License GPL-2

RoxygenNote 5.0.1

NeedsCompilation no

Author Anonymous Author [aut, cre]

Maintainer Anonymous Author <anonymous@author.edu>

R topics documented:

`CrossScreening-package` *Cross-screening for observational studies*

Description

Cross-screening is a new method that uses both random halves of the sample to screen and test many hypotheses. It generally improves statistical power in observational studies when many hypotheses are tested simultaneously.

`bonferroni.fg` *Bonferroni's correction with fixed* Γ

Description

Bonferroni's correction with fixed Γ

Usage

```
bonferroni.fg(d, gamma = 1, mm = c(2, 2, 2), two.sided = TRUE)
```

Arguments

`d`	a matrix of treatment-minus-control differences.
`gamma`	sensitivity parameter (maximum odds different from a randomized experiment).
`mm`	test statistic, either a vector of length 3 or a matrix of three rows where each column corresponds to a U-statistic. Default is the (approximate) Wilcoxon's signed rank test.
`two.sided`	whether a two-sided test should be used. If FALSE, test the one-sided alternative that the center of `d` is positive.

Details

If `mm` is a matrix, this function computes a one-sided or two-sided p-value with each statistic (i.e. there is a p-value for every column of `d` and every column of mm), then does a Bonferroni correction over all the p-values.

Value

a vector of sensitivity values for each column of `d`

`cross.screen` *Cross-screening*

Description

Main functions that implements the cross-screening method in observational studies. `cross.screen` sorts the hypotheses by their sensitivity values and `cross.screen.fg` sorts by p-values at a fixed sensitivity Γ.

Usage

```
cross.screen(d1, d2, gamma = 1, mm = c(2, 2, 2), alpha.screen = 0.05,
  gamma.screen = 0, two.sided = TRUE)

cross.screen.fg(d1, d2, gamma = 1, screen.method = c("threshold",
  "least_sensitive"), alpha.screen = 0.05, alpha.least.sensitive = 2,
  gamma.screen = gamma, mm = c(2, 2, 2), two.sided = TRUE)
```

Arguments

`d1`	screen/test sample (treatment-minus-control differences), can be a matrix (rows are observations, columns are hypotheses)
`d2`	test/screen sample, can be a matrix
`gamma`	sensitivity parameter (maximum odds different from a randomized experiment)
`mm`	a vector of matrix. If matrix, adaptively choose statistic. NULL means Wilcoxon's signed rank statistic.
`alpha.screen`	significance level used in screening.
`gamma.screen`	screening threshold, default is 0, meaning no screening is used.
`two.sided`	if TRUE, automatically select the sign to test; if FALSE, test the one-sided alternative that the center of `d` is positive.
`screen.method`	either keep all hypotheses significant at `gamma.screen` (option "threshold") or keep the least sensitive hypotheses (option "least_sensitive").
`alpha.least.sensitive`	the number of least sensitive hypotheses to keep

Value

`cross.screen` returns a list

s1.kappa kappa values used to screen the hypotheses calculated using the first sample

s1.stat test statistics chosen using the first sample, if `mm` has more than 1 column

s1.side signs of alternative hypotheses chosen using the first sample

s1.order order of the hypotheses by `s1.kappa` if `s1.kappa` is above the threshold `gamma.screen`

p1 p-values computed using the first sample at sensitivity `gamma`

s2.kappa kappa values used to screen the hypotheses calculated using the second sample

s2.stat test statistics chosen using the second sample, if `mm` has more than 1 column

s2.side signs of alternative hypotheses chosen using the second sample

s2.order order of the hypotheses by `s1.kappa` if `s1.kappa` is above the threshold `gamma.screen`

p2 p-values computed using the second sample at sensitivity `gamma`

p Bonferroni adjusted p-values at sensitivity `gamma` computed using `p1` and `p2` (they can be directly used to control FWER)

`cross.screen.fg` returns a list

s1.p p-values used to screen the hypotheses calculated using the first sample

s1.stat test statistics chosen using the first sample, if `mm` has more than 1 column

s1.side signs of alternative hypotheses chosen using the first sample

s1.order order of the hypotheses by `s1.p` if `s1.p` is below the threshold `alpha.screen`

p1 p-values computed using the first sample at sensitivity `gamma`

s2.p p-values used to screen the hypotheses calculated using the second sample

s2.stat test statistics chosen using the second sample, if `mm` has more than 1 column

s2.side signs of alternative hypotheses chosen using the second sample

s2.order order of the hypotheses by `s2.p` if `s2.p` is above the threshold `alpha.screen`

p2 p-values computed using the second sample at sensitivity `gamma`

p Bonferroni adjusted p-values at sensitivity `gamma` computed using `p1` and `p2` (they can be directly used to control FWER)

Functions

- `cross.screen.fg`: Cross-screening with fixed Γ

Examples

```
n <- 100
p <- 20
d <- matrix(rnorm(n * p), n, p)
d[, 1] <- d[, 1] + 2
d1 <- d[1:(n/2), ]
d2 <- d[(n/2+1):n, ]
cross.screen(d1, d2,
             gamma = 9,
             gamma.screen = 1.25)$p

## One can run the hidden function CrossScreening:::table5(no.sims = 1)
## to generate Table 5 in the paper.



## The following code generates Table 1 in the paper.

data(nhanes.fish)
data(nhanes.fish.match)

data <- nhanes.fish
match <- nhanes.fish.match

outcomes <- grep("^o\\.", names(data))
log2diff <- function(y1, y2) {
```

```
    if (min(c(y1, y2)) == 0) {
        y1 <- y1 + 1
        y2 <- y2 + 1
    }
    log2(y1) - log2(y2)
}
d <- sapply(outcomes, function(j) log2diff(data[match$treated, j], data[match$control, j]))
set.seed(11)
split <- sample(1:nrow(d), nrow(d) / 2, replace = FALSE)
d1 <- d[split, ]
d2 <- d[-split, ]

mm <- matrix(c(2, 2, 2, 8, 5, 8), ncol = 2)
data.frame(outcome = names(data)[outcomes],
           p.value =
               cross.screen.fg(d1, d2,
                               gamma = 9,
                               screen.method = "least_sensitive",
                               mm = mm)$p)
```

`fallback.test` *Fallback procedure for multiple testing*

Description

Fallback procedure for multiple testing

Usage

```
fallback.test(p, alpha = 0.05, spread = 1)
```

Arguments

`p`	a vector of p-values
`alpha`	significance level
`spread`	the way to spread `alpha`, either a vector of the same length as `p` or a single number to indicate equal spread in the first `spread` hypotheses.

Value

the rejected hypotheses

References

Brian L. Wiens. A fixed sequence Bonferroni procedure for testing multiple endpoints. Pharmaceutical Statistics, 2(3), 211—215, 2003.

get.kappa *Compute sensitivity value*

Description

Compute sensitivity value

Usage

```
get.kappa(d, alpha = 0.05, mm = c(2, 2, 2))
```

Arguments

d	a vector of treatment-minus-control differences
alpha	significance level
mm	test statistic, either a vector of length 3 or a matrix of three rows where each column corresponds to a U-statistic. Default is the (approximate) Wilcoxon's signed rank test.

Details

The alternative direction is the the center of d is greater than 0.

Value

sensitivity value, i.e. the kappa value such that the p-value becomes just insignificant. If mm is a matrix, then return a vector of sensitivity values corresponding to each column of mm.

nhanes.fish *Health effects of fish*

Description

Data from NHANES (2013-2014) with 1107 observations and 87 variables. Variables whose name start with "o." are lab measurements (such as blood mercury) that can be used as outcomes. The demographics and background variables include gender, age, income, indicator for missing income, race, education, indicator for smoked ever, number of cigararttes smoked in the last month. Fish intakes in the last month (in servings) are summed up in the "fish" variable, which is used to create the binary indicator "fish.level".

Usage

```
data(nhanes.fish)
```

Format

A data.frame.

nhanes.fish.match *Pair matching result*

Description

Each row is a matched pair, the first/second entry is the id of low/high fish intake in the `nhanes.fish` data frame.

Usage

```
data(nhanes.fish.match)
```

Format

A data.frame.

recycle.test *Recycling procedure for multiple testing*

Description

Recycling procedure for multiple testing

Usage

```
recycle.test(p, alpha = 0.05)
```

Arguments

`p`	a vector of p-values
`alpha`	significance level

Details

WARNING: only supports recycle the first two tests.

Value

rejected hypotheses

sen *Sensivity analysis with signed score test*

Description

Sensivity analysis with signed score test

Usage

```
sen(y, mm = NULL, gamma = 1, tail = c("upper", "lower"))
```

Arguments

y	a vector of treatment-minus-control differences
mm	a vector (m, munder, mover) that indicates the U-statistic. NULL means Wilcoxon's signed rank test.
gamma	sensitivity parameter >= 1.
tail	report p-value corresponds to the maximum ("upper") or minimum ("lower") bound

Value

A list

pval p-value

pval2 two sided p-value

T test statistic

E Mean of the test statistic under sensivity gamma

V Variance of the test statistic under sensitivity gamma

devc Effect size of T compared to E and V

Index