Package: jointest (via r-universe)

November 7, 2026

Version 1.0

Date 2024-12-13

Title Multivariate Testing Through Joint Resampling-Based Tests

Description Runs resampling-based tests jointly, e.g., sign-flip score tests from Hemerik et al., (2020) <doi:10.1111/rssb.12369>, to allow for multivariate testing, i.e., weak and strong control of the Familywise Error Rate or True Discovery Proportion.

Imports flipscores, flip, stats, graphics, grDevices

Encoding UTF-8

License GPL-2

RoxygenNote 7.3.3

Suggests knitr, logistf, rmarkdown, ggplot2, lme4

VignetteBuilder knitr

Depends R (>= 2.10)

LazyData true

Repository https://livioivil.r-universe.dev

Date/Publication 2026-06-10 17:50:09 UTC

RemoteUrl https://github.com/livioivil/jointest

RemoteRef HEAD

RemoteSha df8810792780e20e96b6a87e1b468ea5872c7546

Contents

combine-methods 2
flip2sss 3
join_flip2sss 5
join_flipscores 7
jointest-methods 8
oasis 10

Index **11**

combine-methods *Nonparametric combination of* jointest *objects*

Description

Methods for combining jointest objects.

combine_tests combines the tests derived from multiverse models.

combine_contrasts combines the tests derived from the contrasts of a factor variable to get a global test for the factor (i.e. categorical predictor). It has strong analogies with ANOVA test.

Usage

```
combine_tests(mods, comb_funct = "maxT", by = NULL, by_list=NULL, tail = 0)

combine_contrasts(mods, comb_funct = "Mahalanobis", tail = 0)

combine(...)
```

Arguments

mods	a jointest object.
comb_funct	combining function to be used. Several functions are implemented: "mean", "median", "Fisher", "Liptak", (equal to) "Stoufer", "Tippet", (equal to) "minp", "maxT", "Mahalanobis". Alternatively it can be a custom function that has a Tspace matrix as input. For combine_tests the default is comb_funct="maxT", while for combine_contrasts the default is comb_funct="Mahalanobis".
by	if NULL (default), it combines all test statistics. If a characters, it refers to the column's name of summary_table (and printed by something like summary(mods)). The elements with the same value will be combined. If by is a vector, the values are defined by row-wise concatenation of the values of the columns in by. The argument is inactive if by_list is not NULL.
by_list	NULL (default) or a list of vectors. For each vector of the list it combines test statistics with position given by the element of the vector. If the vectors in the list are characters, these refer to names(mods$Tspace).
tail	direction of the alternative hypothesis. It can be "two.sided" (or 0, the default), "less" (or -1) or "greater" (or +1).

Value

The function returns a jointest-object.

Examples

```
#First example
library(jointest)
set.seed(123)

#Simulate data
n=20
D=data.frame(X=rnorm(n),Z1=rnorm(n),Z2=rnorm(n))
D$Y=D$Z1+D$X+rnorm(n)

# Run four glms abd combine it in a list
mod1=glm(Y~X+Z1+Z2,data=D)
mod2=glm(Y~X+poly(Z1,2)+Z2,data=D)
mod3=glm(Y~X+poly(Z1,2)+poly(Z2,2),data=D)
mod4=glm(Y~X+Z1+poly(Z2,2),data=D)
mods=list(mod1=mod1,mod2=mod2,mod3=mod3,mod4=mod4)

# Let us analyze the tests related to coefficient "X" and combine them
res=join_flipscores(mods,n_flips = 5000, seed = 1, tested_coeffs = "X")
summary(combine_tests(res))
# Second (continued) example
# flipscores jointly on all models and all coefficients
res=join_flipscores(mods,n_flips = 2000)
summary(combine_tests(res))
summary(combine_tests(res, by="Model"))
summary(combine_tests(res, by="Coeff"))
res2=combine_contrasts(res)
summary(res2)
#custom combinations:
coeffs=c("(Intercept)","X","Z1","Z2")
coeffs_ids=lapply(coeffs,grep,res2$summary_table$Coeff)
names(coeffs_ids)=coeffs
summary(combine_tests(res2,by_list =   coeffs_ids))
```

`flip2sss` *flipscores 2-Stage Summary Statistics approach*

Description

This function fits a model based on the provided formula and data, accounting for clusters and summary statistics within the model as described in A Andreella, J Goeman, J Hemerik, L Finos (2025) Robust Inference for Generalized Linear Mixed Models: A "Two-Stage Summary Statistics" Approach Based on Score Sign Flipping. Psychometrika, 1-23.

Usage

```
flip2sss(formula = NULL, data = NULL, cluster = NULL,
family = "gaussian", summstats_within=NULL, n_flips = 5000, flips = NULL, ...)
```

Arguments

formula	A formula or a list of formulas. It can be a complete model as.formula or a list of formulas, one for each element produced by the function.
data	The dataset to be used for fitting the model.
cluster	A vector or a formula evaluated on the data that defines the clusters.
family	as in glm, but given as a character. Not used if argument summstats_within is not NULL.
summstats_within	A vector of summary statistics model within the data or a function with argument data.
n_flips	The number of random flips of the score contributions. Overwritten with the nrow(flips) when flips is not NULL (see parameter flips for more details). When n_flips is equal or larger than the maximum number of possible flips (i.e. n^2), all possible flips are performed.
flips	matrix fo +1 or -1, the matrix has n_flips rows and n (number of observations) columns
...	Other arguments passed to the flipscores function.

Value

A jointest object, i.e., a list containing the following objects:

Tspace data.frame where rows represents the sign-flipping transformed (plus the identity one) test and columns the variables.

summary_table data.frame containing for each second-step covariate the estimated parameter, score, std error, test , partial correlation and p-value.

mods List of glm objects, i.e., first-step glm objects

Author(s)

Livio Finos, Angela Andreella

References

Andreella, A., Goeman, J., Hemerik, J., Finos, L. (2025). Robust Inference for Generalized Linear Mixed Models: A "Two-Stage Summary Statistics" Approach Based on Score Sign Flipping. Psychometrika, 1-23. doi: 10.1017/psy.2024.22

See Also

combine_contrasts, combine_tests

Examples

```
library(jointest)
set.seed(123)
# Simulate data
N=20
n=rpois(N,20)
reff=rep(rnorm(N),n)

D=data.frame(X1=rnorm(length(reff)),
             X2=rep(rnorm(N),n),
             Grp=factor(rep(rep(LETTERS[1:3],length.out=N),n)),
             Subj=rep(1:N,n))
D$Y=rbinom(n=nrow(D),prob=plogis( 2*D$X1 * (D$Grp=="B") +  2*D$X2+reff),size=1)

# model of interest formula <- Y ~ Grp * X1 + X2
# clusters structure defined by cluster <- factor(D$Subj)
# The 2-Stage Summary Statistics via flipscore:
res <- flip2sss(Y ~ Grp * X1 + X2, data=D,
               cluster=D$Subj, family="binomial")
summary(res)
# This is an ANOVA-like overall test:
summary(combine_tests(res))
# This is an ANOVA-like test:
summary(combine_contrasts(res))

# An alternative and more flexible definition of the model:
# Define the summary statistics (here we propose the glm with firth correction
# from the logistf package)
summstats_within <- 'logistf::logistf(Y ~ X1, family = binomial(link = "logit"),
control=logistf::logistf.control(maxit=100))'
# however also the classic glm function can be used:
#summstats_within <- 'glm(Y ~ X1, family = binomial(link = "logit"))'

# Then, compute the 2-Stage Summary Statistics approach
# specifying the summary statistics (within cluster/subject)
res <- flip2sss(Y ~ Grp * X1 + X2, data=D, cluster=D$Subj,
                  summstats_within=summstats_within)
summary(res)

# We can also combine the tests:
# Overall:
summary(combine_tests(res))
# This is similar to an ANOVA test:
summary(combine_contrasts(res))
```

`join_flip2sss` *join tests from multiverse flip2sss models*

Description

The function allows hypothesis testing across all plausible multiverse flip2sss models ensuring strong family-wise error rate control.

Usage

```
join_flip2sss(formulas, summstats_within, data, cluster, tested_coeffs = NULL, n_flips = 5000, flips =
```

Arguments

formulas	A formula or a list of formulas. It can be a (list) of complete models as.formula or a list of formulas
summstats_within	A (list of) vector of summary statistics model within the data or a (list of) function with argument data.
data	The dataset to be used for fitting the model.
cluster	A vector or a formula evaluated on the data that defines the clusters.
tested_coeffs	list of the same length of mods, each element of the list being a vector of names of tested coefficients. Alternatively, it can be a vector of names of tested coefficients, in this case, the tested coefficients are attributed to all models (when present). As a last option, it can be NULL, if so, all coefficients are tested.
n_flips	number of flips, default 5000
flips	matrix fo +1 or -1, the matrix has n_flips rows and n (number of observations) columns
seed	NULL by default. Use a number if you wanto to ensure replicability of the results
...	any other further parameter.

Value

A jointest object, i.e., a list containing the following objects:

Tspace data.frame where rows represents the sign-flipping transformed (plus the identity one) test and columns the variables.

summary_table data.frame containing for each model the estimated parameter(s), score(s), std error(s), test(s), partial correlation(s) and p-value(s).

mods List of glms or flipscores objects.

Examples

```
N=20
n=rpois(N,20)
reff=rep(rnorm(N),n)
D=data.frame(X1=rnorm(length(reff)),
               X2=rep(rnorm(N),n),
               Grp=factor(rep(rep(LETTERS[1:3],length.out=N),n)),
               Subj=rep(1:N,n))
D$Y=rbinom(n=nrow(D),prob=plogis( 2*D$X1 * (D$Grp=="B") +  2*D$X2+reff),size=1)
```

```
summstats_within <- list('glm(Y ~ X1, family = binomial(link = "logit"))',
                'logistf::logistf(Y ~ X1^2, family = binomial(link = "logit"), control=logistf::logistf.control(m
formulas <- list(Y ~ Grp * X1 + X2, Y ~ Grp + X1 + X2,  Y ~ Grp + X1)
res <- join_flip2sss(formulas = formulas, summstats_within = summstats_within,
                  data = D, cluster = D$Subj)
```

`join_flipscores` *join tests from multiverse models*

Description

The function allows hypothesis testing across all plausible multiverse models ensuring strong family-wise error rate control.

Usage

```
join_flipscores(mods, tested_coeffs = NULL, n_flips = 5000, flips = NULL,
score_type = "standardized", statistics = "t", seed=NULL, output_models =TRUE, ...)
```

Arguments

`mods`	list of `glms` or `flipscores`-object (or any other object that can be evaluated by `flipscores`)
`tested_coeffs`	list of the same length of `mods`, each element of the list being a vector of names of tested coefficients. Alternatively, it can be a vector of names of tested coefficients, in this case, the tested coefficients are attributed to all models (when present). As a last option, it can be `NULL`, if so, all coefficients are tested.
`n_flips`	number of flips, default 5000
`flips`	matrix fo +1 or -1, the matrix has `n_flips` rows and n (number of observations) columns
`score_type`	any valid type for `flipscores`, `"standardized"` is the default. see `flipscores` for more datails
`statistics`	"t" is the only method implemented (yet). Any other value will not modify the score (a different statistic will only affect the multivariate inference, not the univariate one).
`seed`	`NULL` by default. Use a number if you wanto to ensure replicability of the results
`output_models`	`TRUE` by default. Should the `flipscores` model returned?
`...`	any other further parameter.

Value

A `jointest` object, i.e., a list containing the following objects:

Tspace `data.frame` where rows represents the sign-flipping transformed (plus the identity one) test and columns the variables.

summary_table `data.frame` containing for each model the estimated parameter(s), score(s), std error(s), test(s), partial correlation(s) and p-value(s).

mods List of `glms` or `flipscores` objects.

Examples

```
library(jointest)
set.seed(123)


#EXAMPLE 1: Simulate data:
n=20
D=data.frame(X=rnorm(n),Z1=rnorm(n),Z2=rnorm(n))
D$Y=D$Z1+D$X+rnorm(n)

# Run four glms and combine it in a list
mod1=glm(Y~X+Z1+Z2,data=D)
mod2=glm(Y~X+poly(Z1,2)+Z2,data=D)
mod3=glm(Y~X+poly(Z1,2)+poly(Z2,2),data=D)
mod4=glm(Y~X+Z1+poly(Z2,2),data=D)
mods=list(mod1=mod1,mod2=mod2,mod3=mod3,mod4=mod4)

# flipscores jointly on all models
res=join_flipscores(mods,n_flips = 1000)
summary(combine_tests(res))
summary(combine_tests(res, by="Model"))
summary(combine_contrasts(res))

#Simulate multivariate (50) bionomial responses
set.seed(123)
n=30
D=data.frame(X=rnorm(n),Z=rnorm(n))
Y=replicate(50,rbinom(n,1,plogis(.5*D$Z+.5*D$X)))
colnames(Y)=paste0("Y",1:50)
D=cbind(D,Y)
mods=lapply(1:50,function(i)eval(parse(text=
paste(c("glm(formula(Y",i,"~X+Z),data=D,family='binomial')"),collapse=""))))
# flipscores jointly on all models
res=join_flipscores(mods,n_flips = 1000,tested_coeffs ="X")
summary(res)
res=p.adjust(res)
summary(res)
# Compute lower bound for the true discovery proportion. See packages pARI and sumSome
# install.packages("sumSome")
# install.packages("pARI")
# library(sumSome)
# library(pARI)
# pARI returns a lower bound equals 0.24, i.e., at least 24% of the models
# have a significant effect related to X
# pARI::pARI(ix = c(1:50),pvalues = t(jointest:::.t2p(res$Tspace)),family = "simes",delta = 9)$TDP
# sumSome returns a lower bound equals 0.42, i.e., at least 42% of the models
# have a significant effect related to X
# sumSome::tdp(sumSome::sumStats(G = as.matrix(res$Tspace)))
```

`jointest-methods` *Methods for* `jointest` *objects*

Description

Methods to extract and manipulate relevant information from a `jointest` object.

`print` method for class "`jointest`".

`summary` method for class "`jointest`"

`p.adjust` method for class "`jointest`". Add adjusted p-values into the `jointest` object.

`plot` method for class "`jointest`" This `plot` function visualizes p-values from multiverse models, with different markers to indicate statistical significance levels as defined by the `mark_signif` argument (default is 0.05). Points are plotted with varying shapes based on whether the p-value is below the significance threshold, and colors are used to distinguish between different coefficients.

Usage

```
## S3 method for class 'jointest'
print(x, n = 2, ...)

## S3 method for class 'jointest'
summary(object, digits = 4, ...)

p.adjust(object, method = "maxT", tail = 0, ...)

## S3 method for class 'jointest'
plot(x, ...)
```

Arguments

`x`	an object of class `jointest`.
`...`	additional arguments to be passed, i.e., `mark_signif` and `p.values=c("raw","adjusted")`. See details.
`object`	an object of class `jointest`.
`method`	any method implemented in `flip::flip.adjust` or a custom function. In the last case it must be a function that uses a matrix as input and returns a vector of adjusted p-values equal to the number of columns of the inputed matrix.
`tail`	argument: expresses the tail direction of the alternative hypothesis. It can be "two.sided" (or 0, the default), "less" (or -1) or "greater" (or +1).

Details

`mark_signif` argument: numeric value representing the significance threshold for marking p-values. Any p-value below this threshold will be marked with a dot. The default is `0.05`. `p.values` argument: a character vector specifying which p-values to display. It can be either `"raw"` for raw p-values or `"adjusted"` for adjusted p-values. The default is `"raw"`.

See Also

`flip.adjust`

`oasis` *Longitudinal MRI data in nondemented and demented older adults*

Description

This dataset consists of a longitudinal collection of 150 subjects aged 60 to 96. Each subject was scanned on two or more visits, separated by at least one year for a total of 373 imaging sessions. For each subject, 3 or 4 individual T1-weighted MRI scans obtained in single scan sessions are included.

Usage

```
oasis
```

Format

A data frame with 373 rows and 15 variables:

Subject.ID Subject identification

MRI.ID MRI Exam Identification

Group Class

Visit Visit order

MR.Delay MR Delay Time (Contrast)

Gender Gender

Hand All subjects are right-handed

Age Age of the subject

EDUC Years of Education

SES Socioeconomic Status

MMSE Mini Mental State Examination

CDR Clinical Dementia Rating

eTIV Estimated total intracranial volume

nWBV Normalize Whole Brain Volume

ASF Atlas Scaling Factor

References

https://www.kaggle.com/datasets/jboysen/mri-and-alzheimers

Index

∗ **datasets**
oasis, 10

combine *(*combine-methods*)*, 2
combine-methods, 2
combine_contrasts, *4*
combine_contrasts *(*combine-methods*)*, 2
combine_tests, *4*
combine_tests *(*combine-methods*)*, 2

flip.adjust, *9*
flip2sss, 3
flipscores, *4*, *7*

join_flip2sss, 5
join_flipscores, 7
jointest-methods, 8

oasis, 10

p.adjust *(*jointest-methods*)*, 8
plot.jointest *(*jointest-methods*)*, 8
print.jointest *(*jointest-methods*)*, 8

summary.jointest *(*jointest-methods*)*, 8