

Package: flipscores (via r-universe)

November 3, 2024

Version 1.2.0

Date 2022-08-11

Title Robust Score Testing in GLMs, by Sign-Flip Contributions

Author Livio Finos, Jelle Goeman and Jesse Hemerik, with contribution of Riccardo De Santis.

Maintainer Livio Finos <livio.finos@unipd.it>

Description Provides robust tests for testing in GLMs, by sign-flipping score contributions. The tests are robust against overdispersion, heteroscedasticity and, in some cases, ignored nuisance variables. See Hemerik, Goeman and Finos (2020) <[doi:10.1111/rssb.12369](https://doi.org/10.1111/rssb.12369)>.

Imports MASS, methods

Encoding UTF-8

License GPL-2

RoxygenNote 7.1.2

NeedsCompilation no

Repository CRAN

Date/Publication 2022-08-15 15:30:02 UTC

Contents

| | |
|------------------------------|---|
| flipscores-package | 2 |
| anova.flipscores | 2 |
| compute_scores | 3 |
| flipscores | 4 |
| flipscores-method | 6 |

| | |
|--------------|----------|
| Index | 7 |
|--------------|----------|

flipscores-package *Robust Score Testing in GLMs, by Sign-Flip Contributions*

Description

It provides robust tests for testing in GLMs, by sign-flipping score contributions. The tests are often robust against overdispersion, heteroscedasticity and, in some cases, ignored nuisance variables.

Author(s)

Livio Finos, Jelle Goeman and Jesse Hemerik, with contribution of Riccardo De Santis.

Examples

```
set.seed(1)
dt=data.frame(X=rnorm(20),
  Z=factor(rep(LETTERS[1:3],length.out=20)))
dt$Y=rpois(n=20,lambda=exp(dt$X))
mod=flipscores(Y~Z+X,data=dt,family="poisson",x=TRUE,)
summary(mod)

# Anova test
anova(mod)
# or
mod0=flipscores(Y~Z,data=dt,family="poisson",x=TRUE)
anova(mod0,mod)
# and
mod0=flipscores(Y~X,data=dt,family="poisson")
anova(mod0,mod)
```

anova.flipscores *anova.flipscores*

Description

This is the anova method for flipscores object. Remark: it performs type III deviance decomposition as in `car::Anova`.

Usage

```
## S3 method for class 'flipscores'
anova(object, model1 = NULL, score_type = NULL, n_flips = 5000, id = NULL, ...)
```

Arguments

| | |
|------------|--|
| object | (the object) glm (or flipscores) object with the model under the null hypothesis (i.e. the covariates, the nuisance parameters). |
| model1 | a glm (or flipscores) or a matrix (or vector). If it is a glm object, it has the model under the alternative hypothesis. The variables in model1 are the same variables in object plus one or more variables to be tested. Alternatively, if model1 is a matrix, it contains the tested variables column-wise. |
| score_type | The type of score that is computed. It can be "orthogonalized", "effective" or "basic". Default is "orthogonalized". "effective" and "orthogonalized" take into account the nuisance estimation. The default is NULL, in this case the value is taken from object. |
| n_flips | The number of random flips of the score contributions. When n_flips is equal or larger than the maximum number of possible flips (i.e. n^2), all possible flips are performed. Default is 5000. |
| id | a vector identifying the clustered observations. If NULL (default) observations are assumed to be independent. NOTE: if object is a flipscores and model1\$flip_param_call\$id is not NULL, this is considered in the inference. |
| ... | other parameters allowed in stats::anova. |

Examples

```
set.seed(1)
dt=data.frame(X=scale(rnorm(50)),
              Z=factor(rep(LETTERS[1:3],length.out=50)))
dt$Y=rpois(n=nrow(dt),lambda=exp(dt$X*(dt$Z=="C")))
mod0=flipscores(Y~Z+X,data=dt,family="poisson")
summary(mod0)
anova(mod0)

mod1=flipscores(Y~Z*X,data=dt,family="poisson")
summary(mod1)
anova(mod0,model1 = mod1)
```

compute_scores

compute_scores

Description

Same usage as anova.glm. The parameter id is used too, if present in model0 (with priority) or in model1.

Usage

```
compute_scores(model0, model1, score_type = "standardized")
```

Arguments

| | |
|------------|--|
| model0 | a glm object with the model under the null hypothesis (i.e. the covariates, the nuisance parameters). |
| model1 | a glm or a matrix (or vector). If it is a glm object, it has the model under the alternative hypothesis. The variables in model1 are the same variables in model0 plus one or more variables to be tested. Alternatively, if model1 is a matrix, it contains the tested variables column-wise. |
| score_type | The type of score that is computed. It is "orthogonalized", "effective" or "basic". "effective" and "orthogonalized" take into account the nuisance estimation. |

Author(s)

Jesse Hemerik, Riccardo De Santis, Vittorio Giatti, Jelle Goeman and Livio Finos

Examples

```
set.seed(1)
Z=rnorm(20)
X=Z+rnorm(20)
Y=rpois(n=20,lambda=exp(Z+X))
mod0=glm(Y~Z,family="poisson")
X=data.frame(X=X)
scr0=compute_scores(model0 = mod0, model1 = X)
head(scr0)
```

flipscores

Robust testing in GLMs, by sign-flipping score contributions

Description

Provides robust tests for testing in GLMs, by sign-flipping score contributions. The tests are often robust against overdispersion, heteroscedasticity and, in some cases, ignored nuisance variables.

Usage

```
flipscores(formula, family, data, score_type,
n_flips=5000, alternative = "two.sided",
id = NULL, seed = NULL, ...)
```

Arguments

| | |
|---------|--|
| formula | see glm function. It can also be a model (usually generated by a call to glm); in this case, any other glm-related parameter (e.g. family, data, etc.) are discarded, the function will make use of the ones used to generate the model. |
| family | see glm function. |
| data | see glm function. |

| | |
|-------------|---|
| score_type | The type of score that is computed. It can be "orthogonalized", "effective" or "basic". Both "orthogonalized" and "effective" take into account the nuisance estimation and they provide the same test statistic. In case of small samples "effective score" might have a slight anti-conservative behaviour. "orthogonalized effective score" gives a solution for this issue. Note that in case of a big model matrix, the "orthogonalized" may take a long time. |
| n_flips | The number of random flips of the score contributions. When n_flips is equal or larger than the maximum number of possible flips (i.e. n^2), all possible flips are performed. |
| alternative | It can be "greater", "less" or "two.sided" (default) |
| id | a vector identifying the clustered observations. If NULL (default) observations are assumed to be independent. If id is not NULL, only score_type=="effective" is allowed, yet. |
| seed | NULL by default. |
| ... | see glm function. |

Details

flipscores borrow the same parameters from function glm (and glm.nb). See these helps for more details about parameters such as formula, data, family. Note: in order to use Negative Binomial family, family reference must have quotes (i.e. family="negbinom").

Value

glm class object with sign-flip score test. See also the related functions (summary.flipscores, anova.flipscores, print.flipscores).

Author(s)

Livio Finos, Riccardo De Santis, Vittorio Giatti, Jesse Hemerik and Jelle Goeman

References

"Robust testing in generalized linear models by sign-flipping score contributions" by J.Hemerik, J.Goeman and L.Finos.

See Also

[anova.flipscores](#), [summary.flipscores](#), [flip](#)

Examples

```
set.seed(1)
dt=data.frame(X=rnorm(20),
              Z=factor(rep(LETTERS[1:3],length.out=20)))
dt$Y=rpois(n=20,lambda=exp(dt$Z=="C"))
mod=flipscores(Y~Z+X,data=dt,family="poisson")
summary(mod)
```

```
# Equivalent to:  
model=glm(Y~Z+X,data=dt,family="poisson")  
mod2=flipscores(model)  
summary(mod2)
```

flipscores-method *Methods for flipscores objects*

Description

Methods for flipscores objects. The following are methods to extract and manipulate relevant information from a flipscores object.

Usage

```
## S3 method for class 'flipscores'  
print(x, ...)  
  
## S3 method for class 'flipscores'  
summary(object, ...)
```

Arguments

| | |
|--------|-----------------------------------|
| x | a flipscores object |
| ... | additional arguments to be passed |
| object | a flipscores object |

Index

`anova.flipscores`, [2](#), [5](#)

`compute_scores`, [3](#)

`flip`, [5](#)

`flipscores`, [4](#)

`flipscores-method`, [6](#)

`flipscores-package`, [2](#)

`print.flipscores (flipscores-method)`, [6](#)

`summary.flipscores`, [5](#)

`summary.flipscores (flipscores-method)`,
[6](#)