Package ‘r2glmm’

August 5, 2017

**Type** Package

**Title** Computes R Squared for Mixed (Multilevel) Models

**Date** 2017-08-04

**Version** 0.1.2

**Description** The model R squared and semi-partial R squared for the linear and generalized linear mixed model (LMM and GLMM) are computed with confidence limits. The R squared measure from Edwards et.al (2008) \( <\text{DOI:10.1002/sim.3429} > \) is extended to the GLMM using penalized quasi-likelihood (PQL) estimation (see Jaeger et al. 2016 \( <\text{DOI:10.1080/02664763.2016.1193725} > \)). Three methods of computation are provided and described as follows. First, The Kenward-Roger approach. Due to some inconsistency between the ‘pbkrtest’ package and the ‘glmmPQL’ function, the Kenward-Roger approach in the ‘r2glmm’ package is limited to the LMM. Second, The method introduced by Nakagawa and Schielzeth (2013) \( <\text{DOI:10.1111/j.2041-210x.2012.00261.x} > \) and later extended by Johnson (2014) \( <\text{DOI:10.1111/2041-210X.12225} > \). The ‘r2glmm’ package only computes marginal R squared for the LMM and does not generalize the statistic to the GLMM; however, confidence limits and semi-partial R squared for fixed effects are useful additions. Lastly, an approach using standardized generalized variance (SGV) can be used for covariance model selection. Package installation instructions can be found in the readme file.

**Imports** mgcv, lmerTest, Matrix, pbkrtest, ggplot2, afex, stats, MASS, gridExtra, grid, data.table, dplyr

**Suggests** lme4, nlme, testthat

**License** GPL-2

**LazyData** TRUE

**RoxygenNote** 6.0.1

**URL** https://github.com/bcjaeger/r2glmm

**BugReports** https://github.com/bcjaeger/r2glmm/issues

**NeedsCompilation** no

**Author** Byron Jaeger [aut, cre]
calc_sgv

Compute the standardized generalized variance (SGV) of a blocked diagonal matrix.

Description

Compute the standardized generalized variance (SGV) of a blocked diagonal matrix.

Usage

calc_sgv(nblocks = NULL, blksizes = NULL, vmat)

Arguments

- `nblocks` Number of blocks in the matrix.
- `blksizes` vector of block sizes
- `vmat` The blocked covariance matrix

Value

The SGV of the covariance matrix vmat.

Examples

```r
library(Matrix)
v1 = matrix(c(1, 0.5, 0.5, 1), nrow = 2)
v2 = matrix(c(1, 0.2, 0.1, 0.2, 1, 0.3, 0.1, 0.3, 1), nrow = 3)
v3 = matrix(c(1, 0.1, 0.1, 1, 0.2, 0.1), nrow = 3)
calc_sgv(nblocks = 3, blksizes = c(2, 3, 3), vmat = Matrix::bdiag(v1, v2, v3))
```
**Description**

Compute R2 with a specified C matrix

**Usage**

```r
cmp_R2(c, x, SigHat, beta, method, obsperclust = NULL, nclusts = NULL)
```

**Arguments**

- `c`: Contrast matrix for fixed effects
- `x`: Fixed effects design matrix
- `SigHat`: estimated model covariance (matrix or scalar)
- `beta`: fixed effects estimates
- `method`: the method for computing r2beta
- `obsperclust`: number of observations per cluster (i.e. subject)
- `nclusts`: number of clusters (i.e. subjects)

**Value**

A vector with the Wald statistic (ncp), approximate Wald F statistic (F), numerator degrees of freedom (v1), denominator degrees of freedom (v2), and the specified r squared value (Rsq)

**Examples**

```r
library(nlme)
library(lme4)
library(mgcv)

lmemod = lme(distance ~ age*Sex, random = ~1|Subject, data = Orthodont)
X = model.matrix(lmemod, data = Orthodont)
SigHat = extract.lme.cov(lmemod, data = Orthodont)
beta = fixef(lmemod)
p = length(beta)
obspclust = as.numeric(table(lmemod$data[, 'Subject']))
nclusts = length(obspclust)
C = cbind(rep(0, p-1), diag(p-1))
partial.c = make.partial.C(p-1, p, 2)

cmp_R2(c=c, x=X, SigHat=SigHat, beta=beta, obsperclust = obsperclust, nclusts = nclusts, method = 'sgv')
cmp_R2(c=partial.c, x=X, SigHat=SigHat, beta=beta, obsperclust = obsperclust, nclusts = nclusts, method = 'sgv')
```
glmPQL  

*Compute PQL estimates for fixed effects from a generalized linear model.*

**Description**

Compute PQL estimates for fixed effects from a generalized linear model.

**Usage**

```r
glmPQL(glm.mod, niter = 20, data = NULL)
```

**Arguments**

- `glm.mod`: a generalized linear model fitted with the glm function.
- `niter`: maximum number of iterations allowed in the PQL algorithm.
- `data`: The data used by the fitted model. This argument is required for models with special expressions in their formula, such as offset, log, cbind(successes, trials), etc.

**Value**

A glmPQL object (i.e. a linear model using pseudo outcomes).

**Examples**

```r
# Load the datasets package for example code
library(datasets)
library(dplyr)

# We'll model the number of world changing discoveries per year for the
# last 100 years as a poisson outcome. First, we set up the data
dat = data.frame(discoveries) %>% mutate(year = 1:length(discoveries))

# Fit the GLM with a poisson link function
mod <- glm(discoveries~year+I(year^2), family = 'poisson', data = dat)

# Find PQL estimates using the original GLM
mod.pql = glmPQL(mod)

# Note that the PQL model yields a higher R Squared statistic
# than the fit of a strictly linear model. This is attributed
# to correctly modelling the distribution of outcomes and then
# linearizing the model to measure goodness of fit, rather than
# simply fitting a linear model
summary(mod.pql)
```
is.CompSym

Checks if a matrix is Compound Symmetric.

Description
Checks if a matrix is Compound Symmetric.

Usage
is.CompSym(mat, tol = 1e-05)

Arguments
mat The matrix to be tested.
tol a number indicating the smallest acceptable difference between off diagonal values.

Value
True if the matrix is compound symmetric.

Examples

gcmat <- matrix(c(1,0.2,0.3,0.1,0.2,1,0.2,0.3,0.1,0.3,1,0.1), nrow = 3)
csmat <- matrix(c(1,0.2,0.2,0.2,0.2,0.2,0.2,0.2,0.2), nrow = 3)
is.CompSym(csmat)

make.partial.C

Generate partial contrast matrices

Description
Generate partial contrast matrices

Usage
make.partial.C(rows, cols, index)
Arguments

rows  Number of rows in the contrast matrix
cols  Number of columns in the contrast matrix
index A number corresponding to the position of the fixed effect in the vector of fixed effect parameter estimates.

Value

A contrast matrix designed to test the fixed effect corresponding to index in the vector of fixed effects.

Examples

make.partial.C(4, 5, 2)
make.partial.C(4, 5, 3)
make.partial.C(4, 5, 2:4)

plot.R2  Visualize standardized effect sizes and model R squared

Description

Visualize standardized effect sizes and model R squared

Usage

# S3 method for class 'R2'
plot(x, y = NULL, txtsize = 10, maxcov = 3, r2labs = NULL,
     r2mthd = "sgv", cor = TRUE, ...)

Arguments

x  An R2 object from the r2beta function.
y  An R2 object from the r2beta function.
txtsize The text size of the axis labels.
maxcov Maximum number of covariates to include in the semi-partial plots.
r2labs a character vector containing labels for the models. The labels are printed as subscripts on a covariance model matrix.
r2mthd The method used to compute R2
cor An argument to be passed to the r2dt function. Only relevant if comparing two R2 objects.
... Arguments to be passed to plot
Value

A visual representation of the model and semi-partial R squared from the r2 object provided.

Examples

```r
library(nlme)
library(r2glmm)

data(Orthodont)

# Linear mixed model
lmemod = lme(distance ~ age*Sex, random = ~1|Subject, data = Orthodont)

r2 = r2beta(model=lmemod,partial=TRUE,method='sgv')

plot(x=r2)
```

Description

Fit a GLMM model with multivariate normal random effects using Penalized Quasi-Likelihood for mermod objects.

Usage

```
pqlmer(formula, family, data, niter = 40, verbose = T)
```

Arguments

- `formula`: The lme4 model formula.
- `family`: a family function of the error distribution and link function to be used in the model.
- `data`: the dataframe containing the variables in the model.
- `niter`: Maximum number of iterations to perform.
- `verbose`: if TRUE, iterations are printed to console.

Value

A pseudo linear mixed model of class "lme".

See Also

`glmmPQL`
Examples

# Compare lmer PQL with lme PQL

library(MASS)

lmePQL = glmmPQL(y ~ trt + week + I(week > 2), random = ~ 1 | ID, 
family = binomial, data = bacteria, 
verbose = FALSE)

merPQL = pqlmer(y ~ trt + week + I(week > 2) + (1 | ID), 
family = binomial, data = bacteria, 
verbose = FALSE)

summary(lmePQL)
summary(merPQL)

print.R2

Print the contents of an R2 object

Description
Print the contents of an R2 object

Usage

## S3 method for class 'R2'

print(x, ...)

Arguments

x an object of class R2
...
other arguments passed to the print function.

r2beta

r2beta Compute R Squared for Mixed Models

Description
Computes coefficient of determination (R squared) from edwards et al., 2008 and the generalized 
R squared from Jaeger et al., 2016. Currently implemented for linear mixed models with lmer and 
lme objects. For generalized linear mixed models, only glmmPQL are supported.

Usage

r2beta(model, partial = TRUE, method = "sgv", data = NULL)
**Arguments**

- **model**: a fitted mermod, lme, or glmmPQL model.
- **partial**: if TRUE, semi-partial R squared are calculated for each fixed effect in the mixed model.
- **method**: Specifies the method of computation for R squared beta: if method = 'sgv' then the standardized generalized variance approach is applied. This method is recommended for covariance model selection. if method = 'kr', then the Kenward Roger approach is applied. This option is only available for lme models. if method = 'nsj', then the Nakagawa and Schielzeth approach is applied. This option is available for lmer and lme objects. if method = 'lm', the classical R squared from the linear model is computed. This method should only be used on glm and lm object.
- **data**: The data used by the fitted model. This argument is required for models with special expressions in their formula, such as offset, log, cbind(successes, trials), etc.

**Value**

A dataframe containing the model F statistic, numerator and denominator degrees of freedom, non-centrality parameter, and R squared statistic with 95 If partial = TRUE, then the dataframe also contains partial R squared statistics for all fixed effects in the model.

**References**


**Examples**

```r
library(nlme)
library(lme4)
data(Orthodont)

# Linear mixed models
mermod = lmer(distance ~ age*Sex + (1|Subject), data = Orthodont)
lmemod = lme(distance ~ age*Sex, random = ~1|Subject, data = Orthodont)

# The Kenward-Roger approach
r2beta(mermod, method = 'kr')

# Standardized Generalized Variance
r2beta(mermod, method = 'sgv')
r2beta(lmemod, method = 'sgv')
```
# The marginal R squared by Nakagawa and Schielzeth (extended by Johnson)
r2beta(mermod, method = 'nsj')

# linear and generalized linear models
library(datasets)
dis = data.frame(discoveries)
dis$year = 1:nrow(dis)

lmod = lm(discoveries ~ year + I(year^2), data = dis)
glmod = glm(discoveries ~ year + I(year^2), family = 'poisson', data = dis)

# Using an inappropriate link function (normal) leads to a poor fit relative to the poisson link function.
r2beta(lmod)
r2beta(glm)

# PQL models
# Currently only SGV method is supported
library(MASS)
PQL_bac = glmmPQL(y ~ trt + I(week > 2), random = ~ 1 | ID,
            family = binomial, data = bacteria,
            verbose = FALSE)
r2beta(PQL_bac, method='sgv')

---

r2dt

R Squared Difference Test (R2DT). Test for a statistically significant difference in generalized explained variance between two candidate models.

Description

R Squared Difference Test (R2DT). Test for a statistically significant difference in generalized explained variance between two candidate models.

Usage

r2dt(x, y = NULL, cor = TRUE, fancy = FALSE, onesided = TRUE,
     clim = 95, nsims = 2000, mu = NULL)

Arguments

x An R2 object from the r2beta function.

y An R2 object from the r2beta function. If y is not specified, Ho: E[x] = mu is tested (mu is specified by the user).
cor

if TRUE, the R squared statistics are assumed to be positively correlated and a simulation based approach is used. If FALSE, the R squared are assumed independent and the difference of independent beta distributions is used. This only needs to be specified when two R squared measures are being considered.

fancy

if TRUE, the output values are rounded and changed to characters.

onesided

if TRUE, the alternative hypothesis is that one model explains a larger proportion of generalized variance. If false, the alternative is that the amount of generalized variance explained by the two candidate models is not equal.

clim

Desired confidence level for interval estimates regarding the difference in generalized explained variance.

nsims

number of samples to draw when simulating correlated non-central beta random variables. This parameter is only relevant if cor=TRUE.

mu

Used to test Ho: E[x] = mu.

Value

A confidence interval for the difference in R Squared statistics and a p-value corresponding to the null hypothesis of no difference.

Examples

library(nlme)
library(lme4)
library(r2glm)

data(Orthodont)

# Comparing two linear mixed models
m1 = lmer(distance ~ age*Sex+(1|Subject), Orthodont)
m2 = lmer(distance ~ age*Sex+(1+age|Subject), Orthodont)

m1r2 = r2beta(model=m1,partial=FALSE)
m2r2 = r2beta(model=m2,partial=FALSE)

# Accounting for correlation can make a substantial difference.

r2dt(x=m1r2, y = m2r2, cor = TRUE)
r2dt(x=m1r2, y = m2r2, cor = FALSE)
Index

calc_sgv, 2
cmp_R2, 3

glmmPQL, 7, 8
glmPQL, 4

is.CompSym, 5

lme, 8, 9
lmer, 8, 9

make.partial.C, 5

plot.R2, 6
pqlmer, 7
print.R2, 8

r2beta, 8
r2dt, 10