

Package ‘svylme’

July 23, 2025

Title Linear Mixed Models for Complex Survey Data

Version 1.5-1

Description Linear mixed models for complex survey data, by pairwise composite likelihood, as described in Lumley & Huang (2023) <[doi:10.48550/arXiv.2311.13048](https://doi.org/10.48550/arXiv.2311.13048)>. Supports nested and crossed random effects, and correlated random effects as in genetic models. Allows for multistage sampling and for other designs where pairwise sampling probabilities are specified or can be calculated.

Imports minqa, Matrix, lme4, methods, utils, stats

Depends survey, R (>= 3.5.0)

License GPL-3

Maintainer Thomas Lumley <t.lumley@auckland.ac.nz>

NeedsCompilation no

Author Thomas Lumley [aut, cre]

Repository CRAN

Date/Publication 2024-02-06 16:40:02 UTC

Contents

boot2lme	2
milk_subset	4
nzmaths	5
pisa	7
svy2lme	8
svy2relmer	11
Index	13

boot2lme

*Resampling variances for svy2lme***Description**

Computes variance estimates for the weighted-pairwise-likelihood linear mixed models fitted by [svy2lme](#) using replicate weights. The replicate weights for a pair are obtained by dividing by the sampling weight and then multiplying by the replicate weight. There will be a warning if the ratio of replicate weight to sampling weight differs for observations in the same pair.

Usage

```
boot2lme(model, rdesign, verbose = FALSE)
## S3 method for class 'boot2lme'
vcov(object,
      parameter=c("beta", "theta", "s2", "relSD", "SD", "relVar", "fullVar"),
      ...)
```

Arguments

model	A model returned by svy2lme with the devfun=TRUE option
rdesign	replicate-weights design corresponding to the design used to fit the model, see example
verbose	print progress information?
object	returned by boot2lme
...	for method compatibility
parameter	Variance matrix for: regression parameters, relative variance parameters on Cholesky square root scale, residual variance, relative standard errors of random effects, standard errors of random effects, entire relative variance matrix, entire variance matrix

Details

The variance is estimated from the replicates thetastar and original point estimate theta as $scale \times \sum(rscales \times (thetastar - theta)^2)$.

Value

For boot2lme, an object of class boot2lme with components

theta	replicates of variance parameters (on Cholesky square root scale)
beta	replicates of regression parameters
D	replicates of relative variance matrix
scale, rscales	from the input
formula	model formula from the input

For the vcov method, a variance matrix.

Examples

```

data(api, package="survey")

# two-stage cluster sample
dclus2<-svydesign(id=~dnum+snum, fpc=~fpc1+fpc2, data=apiclus2)

m0<-svy2lme(api00~(1|dnum)+ell+mobility, design=dclus2,return.devfun=TRUE)
jkdes<-as.svrepdesign(dclus2, type="mrb")
jkvar<-boot2lme(m0,jkdes)

SE(jkvar, "beta")
SE(jkvar, "SD")
SE(jkvar,"s2")

m1<-svy2lme(api00~(1|dnum)+ell+mobility,
design=dclus2,return.devfun=TRUE, all.pairs=TRUE, subtract.margins=TRUE)
jk1var<-boot2lme(m1,jkdes)

SE(jk1var, "beta")
SE(jk1var, "SD")

##takes a few minutes
data(pisa)

pisa$w_condstuwt<-with(pisa, w_fstuwt/wnrschbw)
pisa$id_student<-1:nrow(pisa)

dpisa<-survey::svydesign(id=~id_school+id_student, weight=~wnrschbw+w_condstuwt, data=pisa)

m<-svy2lme(isei~(1+female|id_school)+female+high_school+college+one_for+both_for+test_lang,
design=dpisa, return.devfun=TRUE,method="nested")

bpisa<-as.svrepdesign(dpisa, type="bootstrap", replicates=100)

b<-boot2lme(m, bpisa, verbose=TRUE)
str(b)

vcov(b,"beta")
vcov(b,"s2")

## SE() inherits the parameter= argument
SE(b,"beta")
SE(b,"theta")
SE(b,"SD")

```

milk_subset	<i>Milk production (subset)</i>
-------------	---------------------------------

Description

A subset of a dataset from the `pedigreemm` package, created as an example for the `lme4qtl` package. The original data had records of the milk production of 3397 lactations from first through fifty parity Holsteins. These were 1,359 cows, daughters of 38 sires in 57 herds. The data was downloaded from the USDA internet site. All lactation records represent cows with at least 100 days in milk, with an average of 347 days. Milk yield ranged from 4,065 to 19,345 kg estimated for 305 days, averaging 11,636 kg. There were 1,314, 1,006, 640, 334 and 103 records were from first thorough fifth lactation animals. The subset is of cows from 3 sires.

Usage

```
data("milk_subset")
```

Format

A data frame with 316 observations on the following 13 variables.

`id` numeric identifier of cow
`lact` number of lactation for which production is measured
`herd` a factor indicating the herd
`sire` a factor indicating the sire
`dim` number of days in milk for that lactation
`milk` milk production estimated at 305 days
`fat` fat production estimated at 305 days
`prot` protein production estimated at 305 days
`scs` the somatic cell score
`sdMilk` milk scaled by cow-specific standard deviation
`herd_id` a character vector indicating the herd
`one` a vector of 1s for convenience in weighting
`one2` another vector of 1s for convenience in weighting

Details

This data example gives noticeably different results for full likelihood and pairwise likelihood because the model is misspecified. The best fitting linear model for the large herd 89 is different, and that herd gets relatively more weight in the pairwise analysis (because it has more pairs).

Source

Constructed at <https://github.com/variani/lme4qtl/blob/master/vignettes/pedigreemm.Rmd>

References

2010. A.I. Vazquez, D.M. Bates, G.J.M. Rosa, D. Gianola and K.A. Weigel. Technical Note: An R package for fitting generalized linear mixed models in animal breeding. *Journal of Animal Science*, 88:497-504.

Examples

```
data(milk_subset)
herd_des<- svydesign(id = ~herd + id, prob = ~one + one2, data = milk_subset)
lm(sdMilk ~ lact + log(dim),data=milk_subset,subset=herd==89)
lm(sdMilk ~ lact + log(dim),data=milk_subset,subset=herd!=89)
svy2lme(sdMilk ~ lact + log(dim) + (1|herd), design=herd_des,method="nested")
svy2lme(sdMilk ~ lact + log(dim) + (1|herd), design=herd_des,method="general")

## pairwise result is closer to herd 89 than to remainder
lme4::lmer(sdMilk ~ lact + log(dim) + (1|herd), data=milk_subset)
svy2relmer(sdMilk ~ lact + log(dim) + (1|id) + (1|herd), design=herd_des,
  relmat = list(id = A_gen))

## compare to all pairs
svy2lme(sdMilk ~ lact + log(dim) + (1|herd),
  design=herd_des,method="general", all.pairs=TRUE)
svy2lme(sdMilk ~ lact + log(dim) + (1|herd),
  design=herd_des,method="general", all.pairs=TRUE, subtract.margins=TRUE)
```

Description

Data on maths performance, gender, some problem-solving variables and some school resource variables.

Usage

```
data("nzmaths")
```

Format

A data frame with 4291 observations on the following 26 variables.

SCHOOLID School ID

CNT Country id: a factor with levels New Zealand

STRATUM a factor with levels NZL0101 NZL0102 NZL0202 NZL0203

OECD Is the country in the OECD?

STIDSTD Student ID

ST04Q01 Gender: a factor with levels Female Male
 ST14Q02 Mother has university qualifications No Yes
 ST18Q02 Father has university qualifications No Yes
 MATHEFF Mathematics Self-Efficacy: numeric vector
 OPENPS Mathematics Self-Efficacy: numeric vector
 PV1MATH,PV2MATH,PV3MATH,PV4MATH,PV5MATH 'Plausible values' (multiple imputations) for maths performance
 W_FSTUWT Design weight for student
 SC35Q02 Proportion of maths teachers with professional development in maths in past year
 PCGIRLS Proportion of girls at the school
 PROPMA5A Proportion of maths teachers with ISCED 5A (math major)
 ABGMATH Does the school group maths students: a factor with levels No ability grouping between any classes One of these forms of ability grouping between classes for s One of these forms of ability grouping for all classes
 SMRATIO Number of students per maths teacher
 W_FSCHWT Design weight for school
 condwt Design weight for student given school

Source

A subset extracted from the PISA2012lite R package, <https://github.com/pbiecek/PISA2012lite>

References

OECD (2013) PISA 2012 Assessment and Analytical Framework: Mathematics, Reading, Science, Problem Solving and Financial Literacy. OECD Publishing.

Examples

```
data(nzmaths)

oo<-options(survey.lonely.psu="average") ## only one PSU in one of the strata

des<-svydesign(id=~SCHOOLID+STIDSTD, strata=~STRATUM, nest=TRUE,
weights=~W_FSCHWT+condwt, data=nzmaths)

## This example works, but it takes more than five seconds to run, so it
## has been commented out
## m1<-svy2lme(PV1MATH~ (1+ ST04Q01 | SCHOOLID)+ST04Q01*(PCGIRLS+SMRATIO)+MATHEFF+OPENPS, design=des)

options(oo)
```

pisa

Data from the PISA international school survey

Description

Data from the PISA survey of schools, obtained from Stata, who obtained it from Rabe-Hesketh & Skrondal.

Usage

```
data("pisa")
```

Format

A data frame with 2069 observations on the following 11 variables.

female 1 for female

isei socioeconomic index

wfstuwt student sampling weight (total)

wnrschbw school sampling weight

high_school 1 if highest level of parents' education is high school

college 1 if highest level of parents' education is college/uni

one_for 1 if one parent is foreign-born

both_for 1 if both parents are foreign-born

test_lang 1 if the test language is spoken at home

pass_read 1 if the student passed a reading proficiency test

id_school school (sampling unit) identifier

Source

Data downloaded from <https://www.stata-press.com/data/r15/pisa2000.dta>

References

Rabe-Hesketh, S., and A. Skrondal. 2006. Multilevel modelling of complex survey data. *Journal of the Royal Statistical Society, Series A*. 169: 805-827

Examples

```
data(pisa)
```

```
## This model doesn't make a lot of sense, but it's the one in the  
## Stata documentation because the outcome variable is numeric.
```

```
pisa$w_condstuwt<-with(pisa, wfstuwt/wnrschbw)
```

```
pisa$id_student<-1:nrow(pisa)

dpisa<-survey::svydesign(id=~id_school+id_student, weight=~wnrschbw+w_condstwt, data=pisa)

svy2lme(isei~(1|id_school)+female+high_school+college+one_for+both_for+test_lang,
design=dpisa)
```

svy2lme

*Linear mixed models by pairwise likelihood***Description**

Fits linear mixed models to survey data by maximimising the profile pairwise composite likelihood.

Usage

```
svy2lme(formula, design, sterr=TRUE, return.devfun=FALSE,
method=c("general","nested"), all.pairs=FALSE, subtract.margins=FALSE)
## S3 method for class 'svy2lme'
coef(object,...,random=FALSE)
```

Arguments

formula	Model formula as in the lme4 package
design	A survey design object produced by <code>survey::svydesign</code> . The pairwise weights will be computed from this design, which must have separate probabilities or weights for each stage of sampling.
sterr	Estimate standard errors for fixed effects? Set to FALSE for greater speed when using resampling to get standard errors. Also, a PPS-without-replacement survey design can't get sandwich standard errors (because fourth-order sampling probabilities would be needed)
return.devfun	If TRUE, return the deviance function as a component of the object. This will increase the memory use substantially, but allows for bootstrapping.
method	"nested" requires the model clusters to have a single grouping variable that is the same as the primary sampling unit. It's faster.
all.pairs	Only with <code>method="general"</code> , use all pairs rather than just correlated pairs?
subtract.margins	If TRUE and <code>all.pairs=TRUE</code> , compute with all pairs by the faster algorithm involving subtraction from the marginal likelihood
object	svy2lme object
...	for method compatibility
random	if TRUE, the variance components rather than the fixed effects

Details

The population pairwise likelihood would be the sum of the loglikelihoods for a pair of observations, taken over all pairs of observations from the same cluster. This is estimated by taking a weighted sum over pairs in the sample, with the weights being the reciprocals of pairwise sampling probabilities. The advantage over standard weighted pseudolikelihoods is that there is no large-cluster assumption needed and no rescaling of weights. The disadvantage is some loss of efficiency and simpler point estimation.

With `method="nested"` we have the method of Yi et al (2016). Using `method="general"` relaxes the conditions on the design and model.

The code uses `lme4::lmer` to parse the formula and produce starting values, profiles out the fixed effects and residual variance, and then uses `minqa::bobyqa` to maximise the resulting profile deviance.

As with `lme4::lmer`, the default is to estimate the correlations of the random effects, since there is typically no reason to assume these are zero. You can force two random effects to be independent by entering them in separate terms, eg `(1|g)+(-1+x|g)` in the model formula asks for a random intercept and a random slope with no random intercept, which will be uncorrelated.

The internal parametrisation of the variance components uses the Cholesky decomposition of the relative variance matrix (the variance matrix divided by the residual variance), as in `lme4::lmer`. The component `object$s2` contains the estimated residual variance and the component `object$opt$par` contains the elements of the Cholesky factor in column-major order, omitting any elements that are forced to be zero by requiring independent random effects.

Standard errors of the fixed effects are currently estimated using a "with replacement" approximation, valid when the sampling fraction is small and superpopulation (model, process) inference is intended. The influence functions are added up within cluster, centered within strata, the residuals added up within strata, and then the crossproduct is taken within each stratum. The stratum variance is scaled by $n_i/(n_i-1)$ where n_i is the number of PSUs in the stratum, and then added up across strata. When the sampling and model structure are the same, this is the estimator of Yi et al, but it also allows for there to be sampling stages before the stages that are modelled, and for the model and sampling structures to be different.

The `return.devfun=TRUE` option is useful if you want to examine objects that aren't returned as part of the output. For example, `get("ij", environment(object$devfun))` is the set of pairs used in computation.

Value

`svy2lme` returns an object with `print`, `coef`, and `vcov` methods.

The `coef` method with `random=TRUE` returns a two-element list: the first element is the estimated residual variance, the second is the matrix of estimated variances and covariances of the random effects.

Author(s)

Thomas Lumley

References

J.N.K. Rao, François Verret and Mike A. Hidioglou "A weighted composite likelihood approach to inference for two-level models from survey data" Survey Methodology, December 2013 Vol. 39, No. 2, pp. 263-282

Grace Y. Yi , J. N. K. Rao and Haocheng Li "A WEIGHTED COMPOSITE LIKELIHOOD APPROACH FOR ANALYSIS OF SURVEY DATA UNDER TWO-LEVEL MODELS" Statistica Sinica Vol. 26, No. 2 (April 2016), pp. 569-587

Examples

```
data(api, package="survey")

# one-stage cluster sample
dclus1<-svydesign(id=~dnum, weights=~pw, data=apiclus1, fpc=~fpc)
# two-stage cluster sample
dclus2<-svydesign(id=~dnum+snum, fpc=~fpc1+fpc2, data=apiclus2)

svy2lme(api00~(1|dnum)+ell+mobility+api99, design=dclus1)
svy2lme(api00~(1|dnum)+ell+mobility+api99, design=dclus2)
svy2lme(api00~(1|dnum)+ell+mobility+api99, design=dclus2,method="nested")

lme4::lmer(api00~(1|dnum)+ell+mobility+api99, data=apipop)

## Simulated

set.seed(2000-2-29)

df<-data.frame(x=rnorm(1000*20),g=rep(1:1000,each=20), t=rep(1:20,1000), id=1:20000)
df$u<-with(df, rnorm(1000)[g])

df$y<-with(df, x+u+rnorm(1000,s=2))

## oversample extreme `u` to bias random-intercept variance
pg<-exp(abs(df$u/2)-2.2)[df$t==1]

in1<-rbinom(1000,1,pg)==1
in2<-rep(1:5, length(in1))

sdf<-subset(df, (g %in% (1:1000)[in1]) & (t %in% in2))

p1<-rep(pg[in1],each=5)
N2<-rep(20,nrow(sdf))

## Population values
lme4::lmer(y~x+(1|g), data=df)

## Naive estimator: higher intercept variance
lme4::lmer(y~x+(1|g), data=sdf)

##pairwise estimator
sdf$w1<-1/p1
```

```

sdf$w2<-20/5

design<-survey::svydesign(id=~g+id, data=sdf, weights=~w1+w2)
pair<-svy2lme(y~x+(1|g),design=design,method="nested")
pair

pair_g<-svy2lme(y~x+(1|g),design=design,method="general")
pair_g

all.equal(coef(pair), coef(pair_g))
all.equal(SE(pair), SE(pair_g))

```

svy2relmer

*Linear mixed models with correlated random effects***Description**

Fits linear mixed models by maximising the profile pairwise composite likelihood. Allows for correlated random effects, eg, for genetic relatedness (QTL) models

Usage

```
svy2relmer(formula, design, sterr=TRUE, return.devfun=FALSE, relmat=NULL,
  all.pairs=FALSE, subtract.margins=FALSE )
```

Arguments

formula	Model formula as in the lme4 package, or with terms like (1 id) for correlated random effects together with the relmat argument.
design	A survey design object produced by survey::svydesign. The pairwise weights will be computed from this design, which must have separate probabilities or weights for each stage of sampling.
sterr	Estimate standard errors for fixed effects? Set to FALSE for greater speed when using resampling to get standard errors.
return.devfun	If TRUE, return the deviance function as a component of the object. This will increase the memory use substantially, but allows for bootstrapping.
relmat	Specifies a list of relatedness matrices that corresponds to one or more random-effect groupings (eg (1 id) in the formula together with relmat=list(id=Phi) implies a covariance matrix of Phi for the random effects before scaling). See Details and the vignettes.
all.pairs	Use all pairs rather than just correlated pairs?
subtract.margins	If TRUE and all.pairs=TRUE, compute with all pairs by the faster algorithm involving subtraction from the marginal likelihood

Details

This function is very similar to `svy2lme` and only the differences are described here.

Formula parsing and starting values use code based on the `lme4qtl` package.

In `svy2lme` and `lme4::lmer`, the model is based on independent standard Normal random effects that are transformed to give random coefficients that might be correlated within observation but are either identical or independent between observations. In this function, the basic random effects in a term are multiplied by a square root of the `relmat` matrix for that term, giving basic random effects whose covariance between observations proportional to the `relmat` matrix. For example, in a quantitative trait locus model in genetics, the matrix would be a genetic relatedness matrix.

The `relmat` matrices must have dimnames for matching to the `id` variable. It is permissible for the `relmat` matrices to be larger than necessary – eg, containing related units that are not in the sample – since the dimnames will be used to select the relevant submatrix.

There can be only one random-effect term for each `relmat` term. If you need more, make a copy of the term with a different name.

The `return.devfun=TRUE` option is useful if you want to examine objects that aren't returned as part of the output. For example, `get("ij", environment(object$devfun))` is the set of pairs used in computation.

Value

`svy2relmer` returns an object with `print`, `coef`, and `vcov` methods.

Author(s)

Thomas Lumley

References

Ziyatdinov, A., Vázquez-Santiago, M., Brunel, H. et al. `lme4qtl`: linear mixed models with flexible covariance structure for genetic studies of related individuals. *BMC Bioinformatics* 19, 68 (2018). <https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-018-2057-x>

Examples

```
data(milk_subset)
herd_des<- svydesign(id = ~herd + id, prob = ~one + one2, data = milk_subset)

svy2lme(sdMilk ~ lact + log(dim) + (1|herd), design=herd_des, method="general")

svy2relmer(sdMilk ~ lact + log(dim) + (1|id) + (1|herd), design=herd_des,
  relmat = list(id = A_gen))
```

Index

- * **datasets**
 - milk_subset, 4
 - nzmaths, 5
 - pisa, 7
- * **regression**
 - boot2lme, 2
 - svy2lme, 8
 - svy2relmer, 11
- * **survey**
 - boot2lme, 2
 - svy2lme, 8
 - svy2relmer, 11
- A_gen (milk_subset), 4
- boot2lme, 2
- coef.svy2lme (svy2lme), 8
- milk_subset, 4
- nzmaths, 5
- pisa, 7
- svy2lme, 2, 8, 12
- svy2relmer, 11
- vcov.boot2lme (boot2lme), 2