Package 'lmeresampler'

January 31, 2020

Title Bootstrap Methods for Nested Linear Mixed-Effects Models

Version 0.1.1

Description Bootstrap routines for nested linear mixed effects models fit using either 'lme4' or 'nlme'. The provided 'bootstrap()' function implements the parametric, residual, cases, semi-parametric (i.e., CGR), and random effect block (REB) bootstrap procedures. An overview of these procedures can be found in Van der Leeden et al. (2008) <doi: 10.1007/978-0-387-73186-5_11>, Carpenter, Goldstein & Rasbash (2003) <doi: 10.1111/1467-9876.00415>, and Chambers & Chandra (2013) <doi: 10.1080/10618600.2012.681216>.

Depends R (>= 3.1.2)

Imports boot, plyr, dplyr (>= 0.8.0), Matrix, nlmeU, RLRsim

Suggests lme4 (>= 1.1-7), nlme, testthat, mlmRev

License GPL-3

LazyData true

URL https://github.com/aloy/lmeresampler

BugReports https://github.com/aloy/lmeresampler/issues

RoxygenNote 6.1.1

NeedsCompilation no

Author Adam Loy [aut, cre],
Spenser steele [aut]

Maintainer Adam Loy <loyad01@gmail.com>

Repository CRAN

Date/Publication 2020-01-31 17:00:06 UTC

R topics documented:

`bootstrap.lmerMod` *Bootstrap Nested Linear Mixed-Effects Models*

Description

Perform various bootstrap process for nested linear mixed effects (LMEs) models including: parametric, residual, cases, CGR, and REB bootstraps.

Usage

```
## S3 method for class 'lmerMod'
bootstrap(model, fn, type, B, resample, reb_type)

## S3 method for class 'lme'
bootstrap(model, fn, type, B, resample, reb_type)

bootstrap(model, fn, type, B, resample = NULL, reb_type = NULL)
```

Arguments

`model`	The model object you wish to bootstrap.
`fn`	A function returning the statistic(s) of interest.
`type`	A character string indicating the type of bootstrap that is being requested. Possible values are `"parametric"`, `"residual"`, `"case"`, `"cgr"`, or `"reb"` (random effect block bootstrap).
`B`	The number of bootstrap resamples.
`resample`	A logical vector specifying whether each level of the model should be resampled in the cases bootsrap. The levels should be specified from the highest level (largest cluster) of the hierarchy to the lowest (observation-level); for example for students within a school, specify the school level first, then the student level.
`reb_type`	Specification of what random effect block bootstrap version to implement. Possible values are `0`, `1` or `2`.

Details

All of the below methods have been implemented for nested linear mixed-effects models fit by `lmer` (i.e., an `lmerMod` object) and `lme` (i.e., an `lmerMod` object). Details of the bootstrap procedures can be found in the help file for that specific function.

Value

The returned value is an object of class "boot", compatible with the **boot** package's boot methods.

References

Carpenter, J. R., Goldstein, H. and Rasbash, J. (2003) A novel bootstrap procedure for assessing the relationship between class size and achievement. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, **52**, 431–443.

Chambers, R. and Chandra, H. (2013) A random effect block bootstrap for clustered data. *Journal of Computational and Graphical Statistics*, **22**, 452–470.

Morris, J. S. (2002) The BLUPs are not "best" when it comes to bootstrapping. *Statistics and Probability Letters*, **56**, 425–430.

Van der Leeden, R., Meijer, E. and Busing F. M. (2008) Resampling multilevel models. In J. de Leeuw and E. Meijer, editors, *Handbook of Multilevel Analysis*, pages 401–433. New York: Springer.

Bates, D., Maechler, M., Bolker, W., Walker, S. (2015). Fitting Linear Mixed-Effects Models Using lme4. *Journal of Statistical Software*, **67**, 1–48. doi:10.18637/jss.v067.i01.

See Also

- parametric_bootstrap, resid_bootstrap, case_bootstrap, cgr_bootstrap, reb_bootstrap for more details on a specific bootstrap.
- bootMer in the **lme4** package for an implementation of (semi-)parameteric bootstrap for mixed models.
- boot, boot.ci, and plot.boot from the **boot** package.

Examples

```
library(lme4)
vcmodA <- lmer(mathAge11 ~ mathAge8 + gender + class + (1 | school), data = jsp728)

## you can write your own function to return stats, or use something like 'fixef'
mySumm <- function(.) {
  s <- getME(., "sigma")
    c(beta = getME(., "beta"), sigma = s, sig01 = unname(s * getME(., "theta")))
}

## running a parametric bootstrap
set.seed(1234)
boo1 <- bootstrap(model = vcmodA, fn = mySumm, type = "parametric", B = 100)

## Not run:
## running a cases bootstrap - only resampling the schools
boo2 <- bootstrap(model = vcmodA, fn = mySumm, type = "case", B = 100, resample = c(TRUE, FALSE))

## running a cases bootstrap - resampling the schools and students within the school
boo2 <- bootstrap(model = vcmodA, fn = mySumm, type = "case", B = 100, resample = c(TRUE, FALSE))
```

```
## running a semi-parametric bootstrap
boo3 <- bootstrap(model = vcmodA, fn = mySumm, type = "cgr", B = 100)

## running a residual bootstrap
boo4 <- bootstrap(model = vcmodA, fn = mySumm, type = "residual", B = 100)

## running an REB0 bootstrap
boo5 <- bootstrap(model = vcmodA, fn = mySumm, type = "reb", B = 100, reb_typ = 0)

## End(Not run)

## to print results in a formatted way
requireNamespace("boot")
boo1

## you can extract the boostrapped values as a data frame
as.data.frame(boo1$t)

## bootstrap confidence intervals are easily found using 'boot.ci'
##   warnings about "Some ... intervals may be unstable" go away
##   for larger bootstrap samples
boot::boot.ci(boo1, index = 1, type=c("norm", "basic", "perc"))
boot::boot.ci(boo1, index = 6, type=c("norm", "basic", "perc"))

## you can also examine the bootstrap samples graphically
plot(boo1, index = 1)
```

`case_bootstrap.lmerMod` *Cases Bootstrap for Nested LMEs*

Description

Generate cases bootstrap replicates of a statistic for a nested linear mixed-effects model.

Usage

```
## S3 method for class 'lmerMod'
case_bootstrap(model, fn, B, resample)

## S3 method for class 'lme'
case_bootstrap(model, fn, B, resample)

case_bootstrap(model, fn, B, resample)
```

Arguments

`model`	The model object you wish to bootstrap.

`fn`	A function returning the statistic(s) of interest.
`B`	The number of bootstrap resamples.
`resample`	A logical vector specifying whether each level of the model should be resampled in the cases bootsrap. The levels should be specified from the highest level (largest cluster) of the hierarchy to the lowest (observation-level); for example for students within a school, specify the school level first, then the student level.

Details

The cases bootstrap is a fully nonparametric bootstrap that resamples the data with respect to the clusters in order to generate bootstrap samples. Depending on the nature of the data, the resampling can be done only for the higher-level cluster(s), only at the observation-level within a cluster, or at all levels. See Van der Leeden et al. (2008) for a nice discussion of this decision.

To resample a given level of the model, the corresponding entry in the logical vector specified in the `resample` parameter must be set to true. A few examples are given below in terms of a two-level model where students are clustered within schools:

- To resample only the schools, set `resample = c(TRUE,FALSE)`.
- To resample only the students, set `resample = c(FALSE,TRUE)`.
- To resample both the students and the schools, set `resample = c(TRUE,TRUE)`.

Value

The returned value is an object of class "boot", compatible with the **boot** package's `boot` methods.

References

Van der Leeden, R., Meijer, E. and Busing F. M. (2008) Resampling multilevel models. In J. de Leeuw and E. Meijer, editors, *Handbook of Multilevel Analysis*, pages 401–433. New York: Springer.

See Also

- `parametric_bootstrap`, `resid_bootstrap`, `case_bootstrap`, `cgr_bootstrap`, `reb_bootstrap` for more details on a specific bootstrap.
- `bootMer` in the **lme4** package for an implementation of (semi-)parameteric bootstrap for mixed models.
- `boot`, `boot.ci`, and `plot.boot` from the **boot** package.

`cgr_bootstrap.lmerMod` *CGR Bootstrap for Nested LMEs*

Description

Generate semi-parametric bootstrap replicates of a statistic for a nested linear mixed-effects model.

Usage

```
## S3 method for class 'lmerMod'
cgr_bootstrap(model, fn, B)

## S3 method for class 'lme'
cgr_bootstrap(model, fn, B)

cgr_bootstrap(model, fn, B)
```

Arguments

`model`	The model object you wish to bootstrap.
`fn`	A function returning the statistic(s) of interest.
`B`	The number of bootstrap resamples.

Details

The semi-parametric bootstrap algorithm implemented was outlined by Carpenter, Goldstein and Rasbash (2003). The algorithm is outlined below:

1. Obtain the parameter estimates from the fitted model and calculate the estimated error terms and EBLUPs.
2. Rescale the error terms and EBLUPs so that the empirical variance of these quantities is equal to estimated variance components from the model.
3. Sample independently with replacement from the rescaled estimated error terms and rescaled EBLUPs.
4. Obtain bootstrap samples by combining the samples via the fitted model equation.
5. Refit the model and extract the statistic(s) of interest.
6. Repeat steps 3-5 B times.

Value

The returned value is an object of class "boot", compatible with the **boot** package's `boot` methods.

References

Carpenter, J. R., Goldstein, H. and Rasbash, J. (2003) A novel bootstrap procedure for assessing the relationship between class size and achievement. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, **52**, 431–443.

See Also

- parametric_bootstrap, resid_bootstrap, case_bootstrap, cgr_bootstrap, reb_bootstrap for more details on a specific bootstrap.
- bootMer in the **lme4** package for an implementation of (semi-)parameteric bootstrap for mixed models.
- boot, boot.ci, and plot.boot from the **boot** package.

---

jsp728 *Junior school project data*

Description

A dataset containing 728 students from 50 primary (elementary) schools in inner Londan that were part of the Junior School Project (JSP). The variables are as follows:

- mathAge11: score on Maths at age 11
- mathAge8: score on Maths at age 8
- gender: gender, a factor with 2 levels (F, M)
- class: father's social class, a factor with 2 levels (manual, nonmanual)
- shool: school code (1-50)
- normAge11: normalized score on Maths at age 11
- normAge8: normalized score on Maths at age 8
- schoolMathAge8: average school Maths score
- mathAge8c: centered Maths score at age 8

Usage

```
data(jsp728)
```

Format

A data frame with 728 rows and 8 variables.

Source

http://www.bristol.ac.uk/cmm/team/hg/msm-3rd-ed/datasets.html

References

Goldstein, H. (2011). Multilevel Statistical Models (4th ed.). Hoboken: Wiley

`lmeresampler` *Bootstrapping Nested Linear Mixed Effects Models*

Description

The lme4 and nlme packages have made fitting nested linear mixed-effects models quite easy. Using the the functionality of these packages we can easily use maximum likelihood or restricted maximum likelihood to fit our model and conduct inference using our parametric toolkit. In practice, the assumptions of our model are often violated to such a degree that leads to biased estimators and incorrect standard errors. In these situations, resampling methods such as the bootstrap can be used to obtain consistent estimators of the bias and standard errors for inference. lmeresampler provides an easy way to bootstrap nested linear-mixed effects models using either the parametric, residual, cases, CGR (semi-parametric), or random effects block (REB) bootstrap fit using either lme4 or nlme.

`parametric_bootstrap.lmerMod` *Parametric Bootstrap for Nested LMEs*

Description

Generate parametric bootstrap replicates of a statistic for a nested linear mixed-effects model.

Usage

```
## S3 method for class 'lmerMod'
parametric_bootstrap(model, fn, B)

## S3 method for class 'lme'
parametric_bootstrap(model, fn, B)

parametric_bootstrap(model, fn, B)
```

Arguments

`model`	The model object you wish to bootstrap.
`fn`	A function returning the statistic(s) of interest.
`B`	The number of bootstrap resamples.

Details

The parametric bootstrap simulates bootstrap samples from the estimated distribution functions. That is, error terms and random effects are simulated from their estimated normal distributions and are combined into bootstrap samples via the fitted model equation.

Value

The returned value is an object of class "boot", compatible with the **boot** package's `boot` methods.

References

Chambers, R. and Chandra, H. (2013) A random effect block bootstrap for clustered data. *Journal of Computational and Graphical Statistics*, **22**, 452–470.

Van der Leeden, R., Meijer, E. and Busing F. M. (2008) Resampling multilevel models. In J. de Leeuw and E. Meijer, editors, *Handbook of Multilevel Analysis*, pages 401–433. New York: Springer.

See Also

- `parametric_bootstrap`, `resid_bootstrap`, `case_bootstrap`, `cgr_bootstrap`, `reb_bootstrap` for more details on a specific bootstrap.
- `bootMer` in the **lme4** package for an implementation of (semi-)parameteric bootstrap for mixed models.
- `boot`, `boot.ci`, and `plot.boot` from the **boot** package.

`reb_bootstrap.lmerMod` *REB Bootstrap for Two-Level Nested LMEs*

Description

Generate random effect block (REB) bootstrap replicates of a statistic for a two-level nested linear mixed-effects model.

Usage

```
## S3 method for class 'lmerMod'
reb_bootstrap(model, fn, B, reb_type = 0)

## S3 method for class 'lme'
reb_bootstrap(model, fn, B, reb_type = 0)

reb_bootstrap(model, fn, B, reb_type = 0)
```

Arguments

`model`	The model object you wish to bootstrap.
`fn`	A function returning the statistic(s) of interest.
`B`	The number of bootstrap resamples.
`reb_type`	Specification of what random effect block bootstrap version to implement. Possible values are `0`, `1` or `2`.

Details

The random effects block (REB) bootstrap was outlined by Chambers and Chandra (2013) and has been developed for two-level nested linear mixed-effects (LME) models. Consider a two-level LME of the form

$$y = X\beta + Zb + \epsilon$$

The REB bootstrap algorithm (`type = 0`) is as follows:

1. Calculate the nonparametric residual quantities for the fitted model
   - marginal residuals $r = y - X\beta$
   - predicted random effects $\tilde{b} = (Z'Z)^{-1}Z'r$
   - error terms $\tilde{e} = r - Z\tilde{b}$
2. Take a simple random sample with replacement of the groups and extract the corresponding elements of $\tilde{b}$ and $tildee$.
3. Generate bootstrap samples via the fitted model equation $y = X\widehat{\beta} + Z\tilde{b} + \tilde{e}$
4. Refit the model and extract the statistic(s) of interest.
5. Repeat steps 2-4 B times.

Variation 1 (`type = 1`): The first variation of the REB bootstrap zero centers and rescales the residual quantities prior to resampling.

Variation 2 (`type = 2`): The second variation of the REB bootstrap scales the estimates and centers the bootstrap distributions (i.e., adjusts for bias) after REB bootstrapping.

Value

The returned value is an object of class "boot", compatible with the **boot** package's `boot` methods.

References

Chambers, R. and Chandra, H. (2013) A random effect block bootstrap for clustered data. *Journal of Computational and Graphical Statistics*, **22**, 452–470.

See Also

- `parametric_bootstrap`, `resid_bootstrap`, `case_bootstrap`, `cgr_bootstrap`, `reb_bootstrap` for more details on a specific bootstrap.
- `bootMer` in the **lme4** package for an implementation of (semi-)parameteric bootstrap for mixed models.
- `boot`, `boot.ci`, and `plot.boot` from the **boot** package.

`resid_bootstrap.lmerMod` *Residual Bootstrap for Nested LMEs*

Description

Generate residual bootstrap replicates of a statistic for a nested linear mixed-effects model.

Usage

```
## S3 method for class 'lmerMod'
resid_bootstrap(model, fn, B)

## S3 method for class 'lme'
resid_bootstrap(model, fn, B)

resid_bootstrap(model, fn, B)
```

Arguments

`model`	The model object you wish to bootstrap.
`fn`	A function returning the statistic(s) of interest.
`B`	The number of bootstrap resamples.

Details

The residual bootstrap resamples the residual quantities from the fitted linear mixed-effects model in order to generate bootstrap resamples. That is, a random sample, drawn with replacement, is taken from the estimated error terms and the EBLUPS (at each level) and the random samples are combined into bootstrap samples via the fitted model equation.

Value

The returned value is an object of class "boot", compatible with the **boot** package's `boot` methods.

References

Van der Leeden, R., Meijer, E. and Busing F. M. (2008) Resampling multilevel models. In J. de Leeuw and E. Meijer, editors, *Handbook of Multilevel Analysis*, pages 401–433. New York: Springer.

See Also

- `parametric_bootstrap`, `resid_bootstrap`, `case_bootstrap`, `cgr_bootstrap`, `reb_bootstrap` for more details on a specific bootstrap.
- `bootMer` in the **lme4** package for an implementation of (semi-)parameteric bootstrap for mixed models.
- `boot`, `boot.ci`, and `plot.boot` from the **boot** package.

Index