Package 'lmeresampler'

October 13, 2022

Title Bootstrap Methods for Nested Linear Mixed-Effects Models

Version 0.2.2

Description Bootstrap routines for nested linear mixed effects models fit using either 'lme4' or 'nlme'. The provided 'bootstrap()' function implements the parametric, residual, cases, random effect block (REB), and wild 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 (>= 3.5.0)

Imports dplyr (>= 0.8.0), Matrix, nlmeU, ggplot2, ggdist, HLMdiag, purrr, forcats, stats, statmod, tidyr, magrittr, tibble

Suggests lme4 (>= 1.1-7), nlme, testthat, mlmRev, knitr, rmarkdown, doParallel, foreach

License GPL-3

LazyData true

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

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

RoxygenNote 7.1.1

VignetteBuilder knitr

Encoding UTF-8

NeedsCompilation no

Author Adam Loy [aut, cre] (<https://orcid.org/0000-0002-5780-4611>),
Spenser Steele [aut],
Jenna Korobova [aut]

Maintainer Adam Loy <loyad01@gmail.com>

Repository CRAN

Date/Publication 2022-04-29 22:50:11 UTC

R **topics documented:**

---

bootstrap.merMod   *Bootstrap Nested Linear Mixed-Effects Models*

Description

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

Usage

```
## S3 method for class 'merMod'
bootstrap(
  model,
  .f = extract_parameters,
  type,
  B,
  resample,
  reb_type,
  hccme,
  aux.dist,
  orig_data = NULL,
  .refit = TRUE
)

## S3 method for class 'lme'
bootstrap(
```

```
  model,
  .f = extract_parameters,
  type,
  B,
  resample,
  reb_type,
  hccme,
  aux.dist,
  orig_data = NULL,
  .refit = TRUE
)

bootstrap(
  model,
  .f,
  type,
  B,
  resample = NULL,
  reb_type = NULL,
  hccme = NULL,
  aux.dist = NULL,
  orig_data = NULL,
  .refit = TRUE
)
```

Arguments

model	The model object you wish to bootstrap.
.f	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", "wild", 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 bootstrap. 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.
hccme	either "hc2" or "hc3", indicating which heteroscedasticity consistent covariance matrix estimator to use.
aux.dist	one of "mammen", "rademacher", "norm", "webb", or "gamma" indicating which auxiliary distribution to draw the errors from
orig_data	the original data frame. This should be specified if variables are transformed within the formula for glmer() or lmer() and the case bootstrap is used.

.refit   a logical value indicating whether the model should be refit to the bootstrap resample, or if the simulated bootstrap resample should be returned. Defaults to TRUE.

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 "lmeresamp". This is a list with the following elements:

- observed: the estimated values for the model parameters
- model: the fitted model object
- .f: the function call
- replicates: a $B \times p$ data frame of bootstrap values for each of the p model parameters,
- stats: a tibble containing the \]codeobserved, rep.mean (bootstrap mean), se (bootstrap standard error), and bias values for each model parameter,
- B: the number of bootstrap resamples performed
- data: the data with which the model was fit
- seed: a vector of randomly generated seeds that are used by the bootstrap
- type: the type of bootstrap executed
- call: the call to bootstrap() that the user
- message: a list of length B giving any messages generated during refitting. An entry will be NULL if no message was generated.
- warning: a list of length B giving any warnings generated during refitting. An entry will be NULL if no message was generated.
- error: a list of length B giving any errors generated during refitting. An entry will be NULL if no message was generated.

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.

Modugno, L., & Giannerini, S. (2015). The Wild Bootstrap for Multilevel Models. *Communications in Statistics – Theory and Methods*, **44**(22), 4812–4825.

See Also

- parametric_bootstrap, resid_bootstrap, case_bootstrap, reb_bootstrap, wild_bootstrap for more details on a specific bootstrap.

- bootMer in the **lme4** package for an implementation of (semi-)parametric bootstrap for mixed models.

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, .f = mySumm, type = "parametric", B = 20)

## to print results in a formatted way
print(boo1)

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

## running a cases bootstrap - resampling the schools and students within the school
boo3 <- bootstrap(model = vcmodA, .f = mySumm, type = "case", B = 100, resample = c(TRUE, TRUE))

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

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

## Running the Wild bootstrap
boo6 <- bootstrap(model = vcmodA, .f = mySumm, type = "wild", B= 100,
                  hccme = "hc2", aux.dist = "mammen")

## Running a bootstrap in parallel via foreach
library(foreach)
library(doParallel)
set.seed(1234)
```

```
numCores <- 2

cl <- makeCluster(numCores, type = "PSOCK") # make a socket cluster
doParallel::registerDoParallel(cl)          # how the CPU knows to run in parallel

b_parallel <- foreach(B = rep(250, 2),
                      .combine = combine_lmeresamp,
                      .packages = c("lmeresampler", "lme4")) %dopar% {
                        bootstrap(vcmodA, .f = fixef, type = "parametric", B = B)
                      }
stopCluster(cl)

## Running a bootstrap in parallel via parLapply
cl <- makeCluster(numCores, type = "PSOCK") # make a socket cluster
doParallel::registerDoParallel(cl)          # how the CPU knows to run in parallel

boot_mod <- function(...) {
  library(lme4)
  library(lmeresampler)
  vcmodA <- lmer(mathAge11 ~ mathAge8 + gender + class + (1 | school), data = jsp728)
  bootstrap(vcmodA, .f = fixef, type = "parametric", B = 250)

}

result <- parLapply(cl, seq_len(2), boot_mod)
b_parallel2 <- do.call("combine_lmeresamp", result)

stopCluster(cl)

## End(Not run)
```

---

bootstrap_pvals	*Calculate Bootstrap p-values for fixed effects Perform bootstrap tests based on the t-statistic for each fixed effect in order to calculate approximate p-values.*

Description

Calculate Bootstrap p-values for fixed effects

Perform bootstrap tests based on the t-statistic for each fixed effect in order to calculate approximate p-values.

Usage

```
bootstrap_pvals(
  model,
```

```
  type,
  B,
  resample = NULL,
  reb_type = NULL,
  hccme = NULL,
  aux.dist = NULL
)

## S3 method for class 'merMod'
bootstrap_pvals(
  model,
  type,
  B,
  resample = NULL,
  reb_type = NULL,
  hccme = NULL,
  aux.dist = NULL
)

## S3 method for class 'lme'
bootstrap_pvals(
  model,
  type,
  B,
  resample = NULL,
  reb_type = NULL,
  hccme = NULL,
  aux.dist = NULL
)
```

Arguments

model	The model object you wish to bootstrap.
type	A character string indicating the type of bootstrap that is being requested. Possible values are "parametric", "residual", "case", "wild", 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 bootstrap. 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.
hccme	either "hc2" or "hc3", indicating which heteroscedasticity consistent covariance matrix estimator to use.
aux.dist	one of "mammen", "rademacher", "norm", "webb", or "gamma" indicating which auxiliary distribution to draw the errors from

Details

The bootstrap test compares the fitted model specified by the user to reduced models that eliminate a single fixed effect, the same comparison summarized by the table of coefficients in the summary. The bootstrap p-value is then calculated as $(n\_extreme + 1) / (B + 1)$.

Value

A tibble giving the table of coefficients from the model summary with a column appended containing bootstrap p-values.

References

Davison, A., & Hinkley, D. (1997). Tests. In Bootstrap Methods and their Application (Cambridge Series in Statistical and Probabilistic Mathematics, pp. 136-190). Cambridge: Cambridge University Press. doi:10.1017/CBO9780511802843.005

Examples

```
## Not run:
# This takes a while to run
bootstrap_pvals.merMod(jsp_mod, type = "wild", B = 1000, hccme = "hc2", aux.dist = "mammen")

## End(Not run)
```

---

case_bootstrap.merMod   *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 'merMod'
case_bootstrap(model, .f, B, resample, orig_data = NULL, .refit = TRUE)

## S3 method for class 'lme'
case_bootstrap(model, .f, B, resample, orig_data = NULL, .refit = TRUE)

case_bootstrap(model, .f, B, resample, orig_data = NULL, .refit = TRUE)
```

Arguments

model	The model object you wish to bootstrap.
.f	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 bootstrap. 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.

orig_data   the original data frame. This should be specified if variables are transformed within the formula for `glmer()` or `lmer()` and the case bootstrap is used.

.refit   a logical value indicating whether the model should be refit to the bootstrap resample, or if the simulated bootstrap resample should be returned. Defaults to `TRUE`.

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 "lmeresamp".

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

- Examples are given in [bootstrap](#)

- [parametric_bootstrap](#), [resid_bootstrap](#), [case_bootstrap](#), [reb_bootstrap](#), [wild_bootstrap](#) for more details on a specific bootstrap.

- [bootMer](#) in the **lme4** package for an implementation of (semi-)parametric bootstrap for mixed models.

---

combine_lmeresamp   *Combine bootstrap results*

Description

Combines bootstrap results from processes split for parallelization.

Usage

```
combine_lmeresamp(...)
```

Arguments

...   two or more objects of class lmeresamp, to be combined into one.

Details

This helper function combines the different processes split for parallelization to yield unified output and bootstrap statistics.

---

combine_pvals   *Combine p-values*

Description

Combines bootstrap p-values from processes split for parallelization.

Usage

```
combine_pvals(...)
```

Arguments

...   two or more summary data frames produced by bootstrap_pvals.

Details

This helper function combines the different summary tables produced by bootstrap_pvals() when run in parallel to yield unified output and a single summary table.

---

confint.lmeresamp *Calculate confidence intervals for a lmeresamp object*

Description

Calculates normal, basic, and percentile bootstrap confidence intervals from a lmeresamp object.

Usage

```
## S3 method for class 'lmeresamp'
confint(
  object,
  parm,
  level = 0.95,
  type = c("all", "norm", "basic", "perc"),
  ...
)
```

Arguments

object	The lmeresamp object for which confidence intervals should be computed.
parm	not used
level	The level at which the confidence interval should be calculated.
type	A character string giving the type of confidence intervals that should be calculated. This should be a subset of c("norm", "basic", "perc") (for normal, basic, and percentile bootstrap confidence intervals, respectively), or "all".
...	not used

Value

A tibble with columns term, estimate, lower, upper, type, and level.

---

extract_parameters *Extract parameters from LME models*

Description

A utility function that extracts the fixed effects and variance component estimates from a fitted merMod or lme object.

Usage

```
extract_parameters(model)
```

Arguments

model   the fitted model object, either an merMod or lme object.

Value

A named vector of parameters.

---

jsp728   *Junior school project math score data.*

Description

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

Usage

```
data(jsp728)
```

Format

A data frame with 728 rows and 8 variables.

mathAge11 score on math at age 11

mathAge8 score on math at age 8

gender gender, a factor with 2 levels (F, M)

class father's social class, a factor with 2 levels (manual, nonmanual)

school school code (1-50)

normAge11 normalized score on math at age 11

normAge8 normalized score on math at age 8

schoolMathAge8 average school math score

mathAge8c centered math score at age 8

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	*lmeresampler: A package for bootstrapping nested linear mixed-effects models*

Description

The **lme4** and **nlme** packages have made fitting nested linear mixed-effects (LME) models quite easy. Using the the functionality of these packages we can easily use maximum likelihood or restricted maximum likelihood to fit a 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 and standard errors for inference. lmeresampler provides an easy way to bootstrap nested linear-mixed effects models using either fit using either **lme4** or **nlme**.

Details

A variety of bootstrap procedures are available:

- the parametric bootstrap: parametric_bootstrap
- the residual bootstrap: resid_bootstrap
- the cases (i.e. non-parametric) bootstrap: case_bootstrap
- the random effects block (REB) bootstrap: reb_bootstrap
- the Wild bootstrap: wild_bootstrap

In addition to the individual bootstrap functions, lmeresampler provides a unified interface to bootstrapping LME models in its bootstrap function.

---

parametric_bootstrap.merMod

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 'merMod'
parametric_bootstrap(model, .f, B, .refit = TRUE)

## S3 method for class 'lme'
parametric_bootstrap(model, .f, B, .refit = TRUE)

parametric_bootstrap(model, .f, B, .refit = TRUE)
```

Arguments

`model`	The model object you wish to bootstrap.
`.f`	A function returning the statistic(s) of interest.
`B`	The number of bootstrap resamples.
`.refit`	a logical value indicating whether the model should be refit to the bootstrap resample, or if the simulated bootstrap resample should be returned. Defaults to `TRUE`.

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 "lmeresamp".

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

- Examples are given in `bootstrap`
- `parametric_bootstrap`, `resid_bootstrap`, `case_bootstrap`, `reb_bootstrap`, `wild_bootstrap` for more details on a specific bootstrap.
- `bootMer` in the **lme4** package for an implementation of (semi-)parametric bootstrap for mixed models.

---

plot.lmeresamp	*Plot bootstrap results*

Description

Generate a density plot with a half-eye plot representing the 68 percentile intervals from an `lmeresamp` object.

Usage

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

Arguments

x	The lmeresamp object to plot.
var	The estimated parameter to plot, as a string or column number.
...	not used

---

print.coef_tbl	*Print coefficients from a* coef_tbl *object*

Description

Print table of coefficients produced by bootstrap_pvals.

Usage

```
print.coef_tbl(x)
```

Arguments

x	the coef_tbl object to print

---

print.lmeresamp	*Print a summary of an* lmeresamp *object*

Description

Print summary statistics and confidence intervals, if desired, for an lmeresamp object.

Usage

```
## S3 method for class 'lmeresamp'
print(x, ci = FALSE, ...)
```

Arguments

x	The lmeresamp object to print.
ci	A logical value specifying whether confidence intervals should be printed.
...	not used

Details

If the bootstrap statistics are stored in a vector (as opposed to a data frame or tibble), then summary statistics will be calculated and printed. The printed data frame will include the name of the term (if applicable), the observed value (observed), the mean of the bootstrap replicated (rep.mean), the standard error (se), and the bootstrap bias estimate (bias). In addition, the number of resamples will be printed. If any messages, warnings, or errors were generated during the bootstrap procedure, they will be summarized below, and you should check the message, warning, and error elements of the lmeresamp object to investigate further.

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, .f, B, reb_type, .refit = TRUE)

## S3 method for class 'lme'
reb_bootstrap(model, .f, B, reb_type, .refit = TRUE)

reb_bootstrap(model, .f, B, reb_type, .refit = TRUE)
```

Arguments

model	The model object you wish to bootstrap.
.f	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.
.refit	a logical value indicating whether the model should be refit to the bootstrap resample, or if the simulated bootstrap resample should be returned. Defaults to TRUE.

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 predicted random effects, $\tilde{b}$.
3. Draw a simple random sample, with replacement, of the group (cluster) IDs. For each sampled cluster, draw a random sample, with replacement, of size n_i from that cluster's vector of error terms, $\tilde{e}$.

4. Generate bootstrap samples via the fitted model equation $y = X\widehat{\beta} + Z\tilde{b} + \tilde{e}$

5. Refit the model and extract the statistic(s) of interest.

6. Repeat steps 2-5 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 "lmeresamp".

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

- Examples are given in bootstrap

- parametric_bootstrap, resid_bootstrap, case_bootstrap, reb_bootstrap, wild_bootstrap for more details on a specific bootstrap.

- bootMer in the **lme4** package for an implementation of (semi-)parametric bootstrap for mixed models.

---

resid_bootstrap.merMod

Residual Bootstrap for Nested LMEs

Description

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

Usage

```
## S3 method for class 'merMod'
resid_bootstrap(model, .f, B, .refit = TRUE)

## S3 method for class 'lme'
resid_bootstrap(model, .f, B, .refit = TRUE)

resid_bootstrap(model, .f, B, .refit = TRUE)
```

Arguments

`model`	The model object you wish to bootstrap.
`.f`	A function returning the statistic(s) of interest.
`B`	The number of bootstrap resamples.
`.refit`	a logical value indicating whether the model should be refit to the bootstrap resample, or if the simulated bootstrap resample should be returned. Defaults to `TRUE`.

Details

The semi-parametric bootstrap algorithm implemented was outlined by Carpenter, Goldstein and Rasbash (2003), and is referred to as the CGR bootstrap by some. The algorithm is outlined below:

1. Obtain the parameter estimates from the fitted model and calculate the estimated error terms and EBLUPs.

2. Center and 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 "lmeresamp".

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

- Examples are given in bootstrap

- parametric_bootstrap, resid_bootstrap, case_bootstrap, reb_bootstrap, wild_bootstrap for more details on a specific bootstrap.

- bootMer in the **lme4** package for an implementation of (semi-)parametric bootstrap for mixed models.

---

summary.lmeresamp   *Calculate summary statistics for* lmeresamp *objects*

Description

Calculate basic summary statistics such as the mean, standard error, and bias of the bootstrap replicates.

Usage

```
## S3 method for class 'lmeresamp'
summary(object, ...)
```

Arguments

object	The lmeresamp object to be summarized.
...	not used

Details

If the bootstrap statistics are stored in a vector (as opposed to a data frame or tibble), then summary statistics will be calculated. The printed data frame will include the name of the term (if applicable), the observed value (observed), the mean of the bootstrap replicated (rep.mean), the standard error (se), and the bootstrap bias estimate (bias). In addition, the number of resamples will be printed. If any messages, warnings, or errors were generated during the bootstrap procedure, they will be summarized below, and you should check the message, warning, and error elements of the lmeresamp object to investigate further.

---

wild_bootstrap.lmerMod

Wild Bootstrap for LME models

Description

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

Usage

```
## S3 method for class 'lmerMod'
wild_bootstrap(
  model,
  .f,
  B,
  hccme = c("hc2", "hc3"),
  aux.dist = c("mammen", "rademacher", "norm", "webb", "gamma"),
```

```
    .refit = TRUE
)

## S3 method for class 'lme'
wild_bootstrap(
  model,
  .f,
  B,
  hccme = c("hc2", "hc3"),
  aux.dist = c("mammen", "rademacher", "norm", "webb", "gamma"),
  .refit = TRUE
)

wild_bootstrap(model, .f, B, hccme, aux.dist, .refit = TRUE)
```

Arguments

model	The model object you wish to bootstrap.
.f	A function returning the statistic(s) of interest.
B	The number of bootstrap resamples.
hccme	either "hc2" or "hc3", indicating which heteroscedasticity consistent covariance matrix estimator to use.
aux.dist	one of "mammen", "rademacher", "norm", "webb", or "gamma" indicating which auxiliary distribution to draw the errors from
.refit	a logical value indicating whether the model should be refit to the bootstrap resample, or if the simulated bootstrap resample should be returned. Defaults to TRUE.

Details

The wild bootstrap algorithm for LMEs implemented here was outlined by Modugno & Giannerini (2015). The algorithm is outlined below:

1. Draw a random sample equal to the number of groups (clusters) from an auxillary distribution with mean zero and unit variance. Denote these as $w_1, \ldots, w_g$.

2. Calculate the selected heteroscedasticity consistent matrix estimator for the marginal residuals, $\tilde{v}_i$

3. Generate bootstrap responses using the fitted equation: $y_i^* = X_i\beta + \tilde{v}_i w_j$

4. Refit the model and extract the statistic(s) of interest.

5. Repeat steps 2-4 B times.

Value

The returned value is an object of class "lmeresamp".

References

Modugno, L., & Giannerini, S. (2015). The Wild Bootstrap for Multilevel Models. *Communications in Statistics – Theory and Methods*, **44**(22), 4812–4825.

See Also

- Examples are given in `bootstrap`
- `parametric_bootstrap`, `resid_bootstrap`, `case_bootstrap`, `reb_bootstrap`, `wild_bootstrap` for more details on a specific bootstrap.
- `bootMer` in the **lme4** package for an implementation of (semi-)parametric bootstrap for mixed models.

Index