Package 'saeproj.multilevel'

September 7, 2026

Title Small Area Estimation Using a Projection Estimator with a Multilevel Regression Model

Version 0.1.1

Description Provides tools for small area estimation using a projection estimator with a linear multilevel working model. The main function fits a multilevel model to a smaller survey containing the response variable and auxiliary predictors. The fitted model is used to predict outcomes in a larger projection survey, and domain-level estimates are computed by combining synthetic predictions with a design-based residual correction. For methodological references, see Kim and Rao (2012) <doi:10.1093/biomet/asr063>, Food and Agriculture Organization of the United Nations (2021) <doi:10.4060/cb3253en>, and Moura and Holt (1999) <https://www150.statcan.gc.ca/n1/pub/12-001-x/1999001/article/4714-eng.pdf>.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

LazyDataCompression xz

RoxygenNote 8.0.0

Config/roxygen2/version 8.0.0

Depends R (>= 3.5)

Imports cli, dplyr, lme4, reformulas, survey, stats, utils

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

URL https://github.com/rahmanazlya02/saeproj.multilevel

BugReports https://github.com/rahmanazlya02/saeproj.multilevel/issues

NeedsCompilation no

Author Nazlya Rahma Susanto [aut, cre], Azka Ubaidillah [aut]

Maintainer Nazlya Rahma Susanto <susantonazlya@gmail.com>

Repository CRAN

Date/Publication 2026-09-07 18:30:02 UTC

Contents

`as.data.frame.sae_ml_linear` *A Helper Function to Convert sae_ml_linear Results into a Data Frame*

Description

This helper function converts the domain-level estimation results produced by `sae_ml_linear()` into a data frame. The resulting data frame can be used for further analysis, reporting, or visualization.

Usage

```
## S3 method for class 'sae_ml_linear'
as.data.frame(x, row.names = NULL, optional = FALSE, ...)
```

Arguments

`x`	Object of class `"sae_ml_linear"`.
`row.names`	Ignored.
`optional`	Ignored.
`...`	Further arguments.

Value

A data frame containing domain-level estimates.

print.sae_ml_linear — *A Helper Function to Display sae_ml_linear Results, Including the Model Formula, Estimator Type, Number of Domains, and a Preview of Domain-Level Estimates*

Description

This helper function displays the model formula, estimator type, number of domains, and a preview of the domain-level estimates produced by `sae_ml_linear()`.

Usage

```
## S3 method for class 'sae_ml_linear'
print(x, n = 6L, ...)
```

Arguments

`x`	Object of class `"sae_ml_linear"`.
`n`	Number of rows to display.
`...`	Further arguments.

Value

Invisibly returns `x`.

saeml_modelsvy — *An Example Small-Survey Dataset for Fitting the Linear Multilevel Regression Model*

Description

A simulated small-survey dataset used to demonstrate projection-based small area estimation with a linear multilevel regression model.

The dataset is one fixed replication from the simulation design used in the package examples. It contains the target variable and auxiliary variables, and is intended to be used as `data_model` in `sae_ml_linear()`.

Usage

```
saeml_modelsvy
```

Format

A data frame with 250 rows, 11 variables, and 50 domains.

prov Province identifier.

kab_kota District or city identifier used as the domain variable.

id_individu Unique sampled-unit identifier. It is not a PSU or cluster identifier.

Z1 First area-level auxiliary variable.

Z2 Second area-level auxiliary variable.

X1 First unit-level auxiliary variable.

X2 Binary unit-level auxiliary variable.

X3 Third unit-level auxiliary variable.

X4 Fourth unit-level auxiliary variable.

Y Target variable.

WEIND Survey sampling weight.

Details

Simulated model-survey data for saeproj.multilevel.

The dataset contains five sampled units in each of 50 domains. The target variable `Y` was generated from a random-intercept multilevel population model.

It is designed to be used together with `saeml_projsvy`.

The simulated survey design does not include a separate PSU or cluster identifier. Therefore, package examples use `cluster_ids = ~1`.

Source

Simulated data generated from the package study-simulation design.

See Also

`saeml_projsvy` and `sae_ml_linear`.

Examples

```
data(saeml_modelsvy)

dim(saeml_modelsvy)
head(saeml_modelsvy)
table(saeml_modelsvy$kab_kota)
```

`saeml_projsvy` *An Example Large-Survey Dataset for Producing Projection Estimates*

Description

A simulated large-survey dataset used to demonstrate projection-based small area estimation with a linear multilevel regression model.

The dataset is one fixed replication from the simulation design used in the package examples. It contains auxiliary variables and survey design variables, but does not contain the target variable.

It is intended to be used as `data_proj` in `sae_ml_linear()`.

Usage

```
saeml_projsvy
```

Format

A data frame with 15,000 rows, 10 variables, and 50 domains.

prov Province identifier.

kab_kota District or city identifier used as the domain variable.

id_individu Unique sampled-unit identifier. It is not a PSU or cluster identifier.

Z1 First area-level auxiliary variable.

Z2 Second area-level auxiliary variable.

X1 First unit-level auxiliary variable.

X2 Binary unit-level auxiliary variable.

X3 Third unit-level auxiliary variable.

X4 Fourth unit-level auxiliary variable.

WEIND Survey sampling weight.

Details

Simulated projection-survey data for saeproj.multilevel.

The dataset contains 300 sampled units in each of 50 domains. It is drawn from the same fixed simulated population as `saeml_modelsvy`, but does not contain the target variable `Y`.

It is used to generate unit-level predictions and domain-level synthetic estimates.

The simulated survey design does not include a separate PSU or cluster identifier. Therefore, package examples use `cluster_ids = ~1`.

Source

Simulated data generated from the package study-simulation design.

See Also

saeml_modelsvy and sae_ml_linear.

Examples

```
data(saeml_projsvy)

dim(saeml_projsvy)
head(saeml_projsvy)
table(saeml_projsvy$kab_kota)
"Y" %in% names(saeml_projsvy)
```

---

sae_ml_linear *The Main Function of the Package for Small Area Estimation Using a Projection Estimator with a Linear Multilevel Regression Model*

Description

sae_ml_linear() implements a projection estimator for small area estimation using a linear multilevel regression working model fitted with lmer.

The function is designed for situations where information is available from two related surveys:

- **Model survey**: a smaller survey containing the response variable and auxiliary variables.
- **Projection survey**: a larger survey containing auxiliary variables and survey design information, but not the response variable.

The function fits a linear multilevel model using the model survey, predicts the response variable for all units in the projection survey, and produces domain-level projection estimates.

Usage

```
sae_ml_linear(
  formula,
  data_model,
  data_proj,
  domain,
  cluster_ids = ~1,
  weight = NULL,
  strata = NULL,
  summary_function = "mean",
  keep_unit = FALSE,
  seed = 1L,
 control = lme4::lmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 2e+05)),
  return_direct = FALSE,
  ...
)
```

Arguments

formula	A two-sided linear multilevel model formula written using lme4::lmer() syntax. The formula must include at least one random-effect term. For example, Y ~ X1 + X2 + (1 | kab_kota).
data_model	A data frame containing model-survey data. It must contain the response variable, all fixed-effect variables, random-effect grouping variables, domain variables, and survey design variables used in estimation.
data_proj	A data frame containing projection-survey data. It must contain all fixed-effect variables, random-effect grouping variables, domain variables, and survey design variables. The response variable is not required in data_proj.
domain	A character vector or one-sided formula identifying the domain variable or variables used for domain-level aggregation. For example, "kab_kota", c("prov", "kab_kota"), or ~prov + kab_kota.
cluster_ids	A character vector or one-sided formula identifying cluster or primary sampling unit variables used in the survey design. Use ~1 when clustering is not used. The default is ~1.
weight	A character string or one-sided formula identifying the survey weight variable. The specification must identify exactly one variable. Use NULL when sampling weights are not available. The default is NULL.
strata	A character string or one-sided formula identifying the stratification variable used in the survey design. Use NULL when stratification is not used. The default is NULL.
summary_function	Character string specifying the domain-level estimand. Available options are "mean" for domain means and "total" for domain totals. The default is "mean".
keep_unit	Logical. If TRUE, the returned object includes unit-level predictions for data_proj, and unit-level fitted values and model residuals for data_model. The default is FALSE.
seed	Integer value used to set the random seed before model fitting. The default is 1.
control	A control object created by lme4::lmerControl() and passed to lme4::lmer(). The default uses the "bobyqa" optimizer with maxfun = 2e5.
return_direct	Logical. If TRUE, direct design-based estimates are calculated from data_model and included in the returned object. The default is FALSE.
...	Additional arguments passed to survey::svydesign(). For example, nest = TRUE can be supplied for nested cluster designs.

Details

The model formula must follow lme4::lmer() syntax and must include at least one random-effect term.

A random-intercept model can be specified as:

```
Y ~ X1 + X2 + Z1 + Z2 + (1 | kab_kota)
```

where `Y` is the response variable, `X1` and `X2` are unit-level auxiliary variables, `Z1` and `Z2` are auxiliary variables, and `kab_kota` identifies the random-intercept grouping level.

The estimation procedure consists of three main steps:

1. A linear multilevel model is fitted to `data_model` using `lme4::lmer()` with restricted maximum likelihood estimation.
2. Unit-level predictions are generated for all observations in `data_proj`. Predictions use `re.form = NULL` and `allow.new.levels = TRUE`.
3. Predicted values are aggregated by domain to obtain synthetic estimates. A design-based residual correction calculated from `data_model` is added to obtain the final projection estimate.

For domain d, the final projection estimator is:

$$\hat{Y}^{PR} * d = \hat{Y}^{SYN} * d + \hat{B} * d$$

where $\hat{Y}^{SYN} * d$ is the synthetic estimate obtained from the projection survey and $\hat{B} * d$ is the design-based residual correction obtained from the model survey.

The plug-in variance estimator is:

$$\widehat{Var}(\hat{Y}^{PR} * d) = \widehat{Var}(\hat{Y}^{SYN} * d) + \widehat{Var}(\hat{B} * d)$$

The plug-in variance does not account for uncertainty in the estimated multilevel model parameters.

Fixed-effect predictors in `data_proj` must have levels that already exist in `data_model`. New factor levels for fixed-effect predictors produce an error. In contrast, new grouping levels for random effects are allowed and receive a random-effect contribution of zero.

Fixed-effect predictors with zero variance in `data_model` are removed automatically before model fitting. A note identifying removed predictors is stored in the returned object.

If a domain appears in `data_proj` but has no observations in `data_model`, the residual correction and its variance are set to zero. The final estimate for that domain is therefore equal to its synthetic estimate.

Survey design variables, including cluster identifiers, strata, and sampling weights, are used for domain-level aggregation and residual correction through the **survey** package. They are not used as fitting weights in `lme4::lmer()`.

Value

An object of class `"sae_ml_linear"` containing:

call The matched function call.

formula The final model formula used for estimation after removal of any zero-variance fixed-effect predictors.

estimator A character string identifying the estimator as `"bias_corrected"`.

fitted_model The fitted `lmerMod` object returned by `lme4::lmer()`.

model_parameters A list containing fixed effects, random effects, variance components, residual standard deviation, and residual variance.

estimates A data frame containing one row for each domain. It includes domain identifiers, `estimate`, `variance`, `se`, and `rse`.

estimation_details A data frame containing domain identifiers, `estimate_synthetic`, `variance_synthetic`, `correction`, `variance_correction`, `estimate_final`, `variance_final`, `se_final`, `rse_final`, `n_model`, and `n_proj`.

diagnostics A list containing model diagnostics, including residual standard deviation, residual variance, random-effect variance components, intraclass correlation coefficient where applicable, singularity status, convergence information, number of observations, REML status, log-likelihood, AIC, and BIC.

notes A character vector containing notes generated during estimation, such as removed zero-variance predictors, singular fits, convergence issues, negative variances clamped to zero, or domains without residual correction.

unit_projection Returned only when `keep_unit = TRUE`. A data frame containing `data_proj` with an additional `.prediction` column.

unit_model_residual Returned only when `keep_unit = TRUE`. A data frame containing `data_model` with additional `.fitted_model` and `.model_residual` columns.

direct_estimator Returned only when `return_direct = TRUE`. A data frame containing direct design-based estimates, variances, and relative standard errors for each domain.

References

Kim, J. K., & Rao, J. N. K. (2012). Combining data from two independent surveys: A model-assisted approach. *Biometrika*, 99(1), 85–100.

Moura, F. A. S., & Holt, D. (1999). Small area estimation using multilevel models. *Survey Methodology*, 25(1), 73–80.

Bates, D., Maechler, M., Bolker, B., & Walker, S. (2015). Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, 67(1), 1–48.

Food and Agriculture Organization of the United Nations. (2021). *Guidelines on data disaggregation for SDG indicators using survey data* (1st ed.). https://doi.org/10.4060/cb3253en

Finch, W. H., Bolin, J. E., & Kelley, K. (2014). *Multilevel Modeling Using R*. CRC Press.

Hox, J. J., Moerbeek, M., & van de Schoot, R. (2018). *Multilevel Analysis: Techniques and Applications* (3rd ed.). Routledge.

See Also

`lmer` for fitting linear multilevel models, `svydesign` for survey design specification, and `isSingular` for diagnosing singular fits.

Examples

```
data("saeml_modelsvy", package = "saeproj.multilevel")
data("saeml_projsvy", package = "saeproj.multilevel")

result <- sae_ml_linear(
  formula = Y ~ X1 + X2 + X3 + X4 + Z1 + Z2 + (1 | kab_kota),
  data_model = saeml_modelsvy,
```

```
  data_proj = saeml_projsvy,
  domain = "kab_kota",
  cluster_ids = ~1,
  weight = "WEIND",
  strata = "kab_kota",
  summary_function = "mean"
)

summary(result)
```

`summary.sae_ml_linear` *A Helper Function to Display a Summary of sae_ml_linear Results, Including the Model Specification, Estimator Type, Number of Domains, Model Diagnostics, and a Preview of Domain-Level Estimates*

Description

This helper function displays the model specification, estimator type, number of domains, model diagnostics, and a preview of the domain-level estimates produced by `sae_ml_linear()`. The model diagnostics include the number of observations, residual standard deviation, intraclass correlation coefficient (ICC), singular fit status, and convergence status.

Usage

```
## S3 method for class 'sae_ml_linear'
summary(object, n = 6L, ...)
```

Arguments

`object`	Object of class `"sae_ml_linear"`.
`n`	Number of rows to display.
`...`	Further arguments.

Value

Invisibly returns `object`.

Index