Package 'saebenchmarking'

September 28, 2026

Title Benchmarking Small Area Estimates and Their Mean Squared Errors

Version 0.1.0

Maintainer Fiona Audia Nauli Sihombing <fionasihombing95@gmail.com>

Description Adjusts model-based small area estimates so that their weighted aggregate agrees with the weighted aggregate of the direct estimates, using the difference, ratio, and optimum benchmarking methods described in Rao and Molina (2015, ISBN:978-1-118-73578-7) and Wang, Fuller and Qu (2008). The mean squared error (MSE) of the benchmarked empirical best linear unbiased predictor (EBLUP) under the Fay-Herriot model is estimated with the second-order approximation or the parametric bootstrap of Steorts and Ghosh (2013) <doi:10.5705/ss.2012.053>. The posterior MSE of the benchmarked hierarchical Bayes (HB) estimator follows Datta, Ghosh, Steorts and Maples (2011) <doi:10.1007/s11749-010-0218-y>.

License MIT + file LICENSE

URL https://github.com/fionaaudia/saebenchmarking

BugReports https://github.com/fionaaudia/saebenchmarking/issues

Depends R (>= 3.5)

Imports sae, stats, withr

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

Language en-US

LazyData true

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author Fiona Audia Nauli Sihombing [aut, cre, cph], Azka Ubaidillah [aut]

Repository CRAN

Date/Publication 2026-09-28 08:00:02 UTC

Contents

`data_eblup` *Sample Data for Fay-Herriot Model with EBLUP Estimation*

Description

Dataset to simulate benchmarking of small area estimates obtained by the empirical best linear unbiased predictor (EBLUP) under the Fay-Herriot model.

This data is generated based on the Fay-Herriot model by these following steps:

1. Take the population size `pop` (in thousands) and the sample size `n` of 50 areas from Table 1 of Wang, Fuller and Qu (2008).
2. Calculate the known sampling variance $\psi_i = 16/n_i$ and the benchmarking weight $w_i = pop_i / \sum pop_i$.
3. Generate the auxiliary variable `z ~ N(1, 1)` once and keep it fixed.
4. Generate the random effect `v ~ N(0, 1)`. Set β_0 = 6 and β_1 = 3, then calculate the true area mean $\theta_i = \beta_0 + \beta_1 z_i + v_i$.
5. For each area, generate n_i unit observations $\theta_i + \varepsilon_{ij}$ with $\varepsilon_{ij} \sim N(0, 16)$. Calculate the direct estimation `direct` as their mean and the sampling error as `e = direct - theta`.
6. Estimate the random effect variance by restricted maximum likelihood (REML), then calculate the EBLUP `eblup` and its mean squared error `mse` using `sae::mseFH()`.
7. Combine all variables in a data frame named `data_eblup`. The REML estimate of the random effect variance is stored as the attribute `sigma2_v`.

Usage

```
data(data_eblup)
```

Format

A data frame with 50 rows and 12 variables:

area Area identifier, 1 to 50

pop Population size of each area (in thousands)

n Sample size of each area

weight Benchmarking weight, `pop / sum(pop)`

z Auxiliary variable

v True random effect

e True sampling error

theta True small area mean

direct Direct estimation

vardir Known sampling variance, `16 / n`

eblup EBLUP estimation

mse Mean squared error of the EBLUP estimation

Details

The attribute `sigma2_v` is dropped when the data frame is subset. Read it first with `attr(data_eblup, "sigma2_v")`.

References

Molina, I. and Marhuenda, Y. (2015). sae: An R package for small area estimation. *The R Journal*, 7(1), 81-98.

Wang, J., Fuller, W. A. and Qu, Y. (2008). Small area estimation under a restriction. *Survey Methodology*, 34(1), 29-36.

You, Y., Rao, J. N. K. and Hidiroglou, M. (2013). On the performance of self benchmarked small area estimators under the Fay-Herriot area level model. *Survey Methodology*, 39(1), 217-229.

`data_hb` *Sample Data for Fay-Herriot Model with Hierarchical Bayes Estimation*

Description

Dataset to simulate benchmarking of small area estimates obtained by the hierarchical Bayes (HB) method under the Fay-Herriot model.

This data is generated based on the Fay-Herriot model by these following steps:

1. Generate `pop`, `n`, `weight`, `z`, `v`, `e`, `theta`, `direct`, and `vardir` by steps 1 to 5 of `data_eblup`. Both datasets share the same values of these variables.
2. Set the uniform prior $\pi(\beta, \tau^2) \propto 1$ on the regression coefficients and the random effect variance τ^2, as in Sugasawa, Tamae and Kubokawa (2017), and treat the sampling variance `vardir` as known.
3. Run the Gibbs sampler by updating in turn τ^2 from the inverse gamma distribution, β from the bivariate normal distribution, and θ_i from the normal distribution, using their full conditional distributions.
4. Discard the first 1,000 iterations as burn-in and keep the next 5,000 draws.
5. Calculate the HB estimation `theta_hb` as the mean of the kept draws and the posterior variance `var_hb` as their variance.
6. Combine all variables in a data frame named `data_hb`.

Usage

```
data(data_hb)
```

Format

A data frame with 50 rows and 12 variables:

area Area identifier, 1 to 50

pop Population size of each area (in thousands)

n Sample size of each area

weight Benchmarking weight, `pop / sum(pop)`

z Auxiliary variable

v True random effect

e True sampling error

theta True small area mean

direct Direct estimation

vardir Known sampling variance, `16 / n`

theta_hb HB estimation (posterior mean)

var_hb Posterior variance of the HB estimation

Details

Convergence was checked by comparing three chains with different starting values and update orders.

References

Sugasawa, S., Tamae, H. and Kubokawa, T. (2017). Bayesian estimators for small area models shrinking both means and variances. *Scandinavian Journal of Statistics*, 44(1), 150-167. doi:10.1111/sjos.12246

Wang, J., Fuller, W. A. and Qu, Y. (2008). Small area estimation under a restriction. *Survey Methodology*, 34(1), 29-36.

You, Y., Rao, J. N. K. and Hidiroglou, M. (2013). On the performance of self benchmarked small area estimators under the Fay-Herriot area level model. *Survey Methodology*, 39(1), 217-229.

mse_benchmarking *Estimate the MSE of Benchmarked Small Area Estimates*

Description

Computes the mean squared error (MSE) of small area estimates benchmarked with sae_benchmarking(), for the ratio, difference, or optimum method, when the original estimates are EBLUPs under the Fay-Herriot model or hierarchical Bayes (HB) estimates.

Usage

```
mse_benchmarking(
  object,
  estimator = c("eblup", "hb"),
  z = NULL,
  sigma2_v = NULL,
  fitting_method = c("REML", "PR"),
  theta_hb = NULL,
  V_hb = NULL,
  posterior_draws = NULL,
  B = 1000,
  seed = NULL
)
```

Arguments

object	An object of class "sae_benchmarking" returned by sae_benchmarking(). vardir must have been supplied in that call when estimator = "eblup".
estimator	Character. "eblup" or "hb".
z	Design matrix (areas in rows, including the intercept column if any) used to fit the Fay-Herriot model. Required when estimator = "eblup".
sigma2_v	Estimated random effect variance from the original Fay-Herriot fit. Required when estimator = "eblup".
fitting_method	Character. "REML" or "PR" (Prasad-Rao moment method), matching how sigma2_v was estimated. Used only for method = "difference" under estimator = "eblup".
theta_hb	Numeric vector of HB estimates (posterior means) in the same order as the areas of object. Used when estimator = "hb".
V_hb	Numeric vector of HB posterior variances in the same order as the areas of object. Used when estimator = "hb".
posterior_draws	Matrix of MCMC draws of the unbenchmarked small area means (areas in rows, draws in columns). Used when estimator = "hb" and theta_hb or V_hb is not supplied.
B	Positive integer. Number of bootstrap replicates, used for the ratio and optimum methods under estimator = "eblup".

seed — Optional integer seed for the bootstrap. The random number generator state of the session is restored on exit.

Value

A list of class "mse_benchmarking" with elements:

- method: the benchmarking method of object.
- estimator: "eblup" or "hb".
- estimate: named numeric vector of benchmarked estimates.
- mse: named numeric vector of estimated MSEs.
- B: number of bootstrap replicates requested, or NA when no bootstrap was used.
- n_failed: number of discarded bootstrap replicates, or NA when no bootstrap was used.

See Also

sae_benchmarking()

Examples

```
# EBLUP, difference benchmarking (closed form)
bm_diff <- sae_benchmarking(method = "difference", direct = "direct",
                            weight = "weight", estimate = "eblup",
                            mse = "mse", vardir = "vardir",
                            data = data_eblup)
Z <- cbind(1, data_eblup$z)
s2v <- attr(data_eblup, "sigma2_v")
res_diff <- mse_benchmarking(bm_diff, estimator = "eblup", z = Z,
                             sigma2_v = s2v, fitting_method = "REML")
head(res_diff$mse)
```

---

sae_benchmarking *Benchmark small area estimates*

Description

Applies difference, ratio, or optimum benchmarking to small area estimates so that their weighted aggregate equals the weighted aggregate of the direct estimates.

Usage

```
sae_benchmarking(
  method = c("ratio", "difference", "optimum"),
  direct,
  weight,
  estimate,
```

```
  mse = NULL,
  vardir = NULL,
  phi_source = c("mse", "vardir"),
  area_names = NULL,
  data = NULL
)
```

Arguments

method	Character. One of "ratio", "difference", or "optimum".
direct	Numeric vector of direct estimates, or the name of a column of data. Used to compute the target sum(weight * direct).
weight	Numeric vector of non-negative benchmarking weights (e.g. population proportions), or the name of a column of data. Rescaled to sum to one.
estimate	Numeric vector of model-based small area estimates (e.g. EBLUP or HB), or the name of a column of data.
mse	Optional numeric vector of mean squared errors of estimate, or the name of a column of data. Required when method = "optimum" and phi_source = "mse".
vardir	Optional numeric vector of sampling variances of direct, or the name of a column of data. Required when method = "optimum" and phi_source = "vardir", and by mse_benchmarking() with estimator = "eblup".
phi_source	Character. One of "mse" or "vardir"; the quantity used to build the optimum benchmarking factor. Ignored for other methods.
area_names	Optional character vector of area labels. Defaults to "Area_1", "Area_2", and so on.
data	Optional data frame. When supplied, direct, weight, estimate, mse, and vardir may be given as column names.

Value

This function returns a list of class "sae_benchmarking" with elements:

- method: the benchmarking method used.
- target: numeric value of the target.
- estimates: a one-column data frame (Estimate) of benchmarked estimates, with area names as row names.
- params: a list with the adjustment parameter (adjustment, ratio, or lambda, depending on method).
- aggregation: a one-row matrix with columns Direct, Estimate, Bench, and Target giving the weighted aggregates.
- inputs: a list of the inputs used, retained for mse_benchmarking().

See Also

mse_benchmarking() to estimate the mean squared error of the benchmarked estimates.

Examples

```
est    <- c(10.2, 8.7, 12.1)
direct <- c(10.5, 8.2, 12.8)
mse    <- c(0.5, 0.4, 0.6)
w      <- c(0.3, 0.4, 0.3)

res <- sae_benchmarking(method = "difference", direct = direct,
                        weight = w, estimate = est, mse = mse)
res
coef(res)
summary(res)
```

Index