Package 'interSAE'

August 25, 2026

Type Package

Title Intersectional Small Area Estimation from Survey and Census Data

Version 0.1.0

Description Implements design-aware, margin-constrained estimation of population indicators for geographic areas crossed with demographic subgroups. Complex-survey microdata supply outcomes and association information, while one or more aggregate census tables supply overlapping population margins. A generalized iterative proportional fitting engine reconstructs a coherent latent population table, and an augmented model-assisted estimator produces domain means, proportions, and totals for sampled and unsampled intersections. Tools diagnose non-identification, compute linear-programming sensitivity bounds, propagate sampling and model uncertainty with replicate-weight or multiplier bootstrap procedures, enforce structural zeros, and benchmark estimates to official totals. The framework extends calibration ideas from Deville and Sarndal (1992) <doi:10.1080/01621459.1992.10475217> and small area estimation ideas from Fay and Herriot (1979) <doi:10.1080/01621459.1979.10482505>.

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 4.1.0)

Imports graphics, stats, utils

Suggests knitr, lpSolve, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

RoxygenNote 7.3.2

NeedsCompilation no

Author Leila Marvian Mashhad [aut, cre]

Maintainer Leila Marvian Mashhad <leila.marveian@gmail.com>

Repository CRAN

Date/Publication 2026-08-25 05:20:31 UTC

Contents

`interSAE-package` *Intersectional Small Area Estimation from Survey and Census Data*

Description

The **interSAE** package combines complex-survey microdata with overlapping aggregate census margins. It reconstructs a coherent latent population table and produces model-assisted estimates for geographic areas crossed with demographic subgroups.

Details

The package separates survey-design declaration, census-margin validation, constrained population reconstruction, and outcome estimation. Optional tools diagnose non-identification, calculate association-sensitivity bounds, bootstrap the complete fitting procedure, and benchmark estimates.

Author(s)

Leila Marvian Mashhad

References

Deville, J.-C. and Sarndal, C.-E. (1992). Calibration estimators in survey sampling. *Journal of the American Statistical Association*, 87, 376–382. doi:10.1080/01621459.1992.10475217.

Fay, R. E. and Herriot, R. A. (1979). Estimates of income for small places. *Journal of the American Statistical Association*, 74, 269–277. doi:10.1080/01621459.1979.10482505.

benchmark_inter *Benchmark Domain Estimates to an Official Aggregate*

Description

Apply a common adjustment while reproducing an official population-weighted aggregate. Logit adjustment preserves a bounded response range.

Usage

```
benchmark_inter(estimates, target, weight = "population",
  method = c("logit", "ratio", "additive"), lower = 0, upper = 1)
```

Arguments

estimates	A fitted inter_sae object or estimate data frame.
target	Official weighted mean or proportion.
weight	Benchmarking-weight column.
method	Logit, ratio, or additive adjustment.
lower, upper	Range for bounded logit adjustment.

Value

A data frame with original and benchmarked estimates.

Examples

```
example_data <- inter_example()
design <- inter_design(example_data$survey, "weight", "area")
margins <- census_margins(example_data$margins)
fit <- fit_inter_sae(unemployed ~ sex + age + residence + area,
  design, margins, family = "binomial")
official <- with(example_data$survey,
  weighted.mean(unemployed, weight))
benchmarked <- benchmark_inter(fit, official)
attr(benchmarked, "benchmark_achieved")
```

`fit_inter_sae` *Fit and Inspect Intersectional Small Area Estimates*

Description

Reconstruct a coherent latent joint population table, fit a survey-weighted outcome model, and calculate augmented model-assisted estimates.

Usage

```
fit_inter_sae(formula, design, margins,
  family = c("gaussian", "binomial"), domain = NULL, prior = 0.5,
  max_iter = 1000L, tol = 1e-8, residual_correction = TRUE,
  max_cells = 1e6, model_args = list())

inter_estimates(object, statistic = c("mean", "total", "both"))

margin_diagnostics(object)

## S3 method for class 'inter_sae'
print(x, ...)

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

## S3 method for class 'summary_inter_sae'
print(x, ...)

## S3 method for class 'inter_sae'
plot(x, y = NULL, compare_direct = TRUE, ...)
```

Arguments

`formula`	Outcome model formula. Every predictor must occur among the census-margin dimensions.
`design`	An object returned by `inter_design()`.
`margins`	An object returned by `census_margins()`.
`family`	Gaussian or binomial response family.
`domain`	Character vector of output-domain dimensions.
`prior`	Positive seed added to every feasible latent population cell.
`max_iter`	Maximum generalized iterative-proportional-fitting iterations.
`tol`	Relative convergence tolerance.
`residual_correction`	Add a calibrated survey-residual correction.

max_cells	Maximum permitted latent-table size.
model_args	Optional arguments passed to stats::glm().
object, x	A fitted inter_sae object.
statistic	Extract means/proportions, totals, or both.
y	Ignored.
compare_direct	Overlay direct survey estimates in the plot.
...	Additional printing or plotting arguments.

Details

For binomial outcomes the working model uses a quasibinomial likelihood so non-integer survey weights are interpreted as analytic weights. Returned estimates remain proportions on the response scale.

Value

fit_inter_sae() returns an inter_sae object. inter_estimates() and margin_diagnostics() return data frames. The summary method returns a summary_inter_sae object. The plot method returns the plotted data invisibly.

Examples

```
example_data <- inter_example()
design <- inter_design(example_data$survey, "weight", "area",
  "strata", "psu")
margins <- census_margins(example_data$margins)
fit <- fit_inter_sae(unemployed ~ sex + age + residence + area,
  design, margins, family = "binomial", domain = c("area", "sex"))
head(inter_estimates(fit))
max(margin_diagnostics(fit)$relative_error)
summary(fit)
```

---

identification_bounds *Diagnose Non-Identification and Compute Sensitivity Bounds*

Description

Measure the nullity left by overlapping aggregate margins and, when **lpSolve** is installed, optimize the target over latent population tables that satisfy the margins and optional association restrictions.

Usage

```
check_identifiability(object, tol = 1e-9)

identification_bounds(object, domain = NULL, gamma = Inf,
  statistic = c("mean", "total"))

inter_sensitivity(object, gamma = c(1, 1.25, 1.5, 2, Inf),
  domain = NULL, statistic = c("mean", "total"))

## S3 method for class 'inter_identifiability'
print(x, ...)

## S3 method for class 'inter_bounds'
print(x, ...)
```

Arguments

object	A fitted inter_sae object.
tol	Numerical-rank tolerance.
domain	Optional named list defining one target domain.
gamma	Association-sensitivity value or vector. One fixes the fitted latent table; infinity imposes census margins only.
statistic	Bound a predicted domain mean/proportion or total.
x	Object to print.
...	Unused.

Details

Mean bounds use a Charnes–Cooper transformation of the linear-fractional target. They concern uncertainty in the latent census table conditional on the fitted cell predictions; they are not replacements for sampling intervals.

Value

check_identifiability() returns an inter_identifiability object. identification_bounds() returns an inter_bounds object. inter_sensitivity() returns a data frame.

Examples

```
example_data <- inter_example()
design <- inter_design(example_data$survey, "weight", "area")
margins <- census_margins(example_data$margins)
fit <- fit_inter_sae(unemployed ~ sex + age + residence + area,
  design, margins, family = "binomial")
check_identifiability(fit)
if (requireNamespace("lpSolve", quietly = TRUE)) {
  identification_bounds(fit, domain = list(area = "A1"), gamma = 1.5)
}
```

`inter_bootstrap` *Bootstrap and Summarize Uncertainty*

Description

Refit population reconstruction and the outcome model under each set of weights. Standard errors include sampling and refitting variation. Identification width can be reported separately.

Usage

```
inter_bootstrap(object, R = 199L,
  type = c("multiplier", "replicate"), seed = 123,
  level = 0.95, progress = FALSE)

decompose_uncertainty(object, bootstrap = NULL, bounds = NULL)

## S3 method for class 'inter_bootstrap'
print(x, ...)
```

Arguments

`object`	A fitted `inter_sae` object.
`R`	Number of bootstrap replicates.
`type`	Multiplier or supplied-replicate-weight bootstrap.
`seed`	Random seed for multiplier bootstrap.
`level`	Confidence level.
`progress`	Print periodic progress messages.
`bootstrap`	Optional result from `inter_bootstrap()`.
`bounds`	Optional result from `identification_bounds()`.
`x`	Object to print.
`...`	Unused.

Value

`inter_bootstrap()` returns an `inter_bootstrap` object containing a summary and replicate matrix. `decompose_uncertainty()` returns a data frame.

Examples

```
## Not run:
example_data <- inter_example()
design <- inter_design(example_data$survey, "weight", "area", "strata", "psu")
margins <- census_margins(example_data$margins)
fit <- fit_inter_sae(unemployed ~ sex + age + residence + area,
  design, margins, family = "binomial")
```

```
boot <- inter_bootstrap(fit, R = 99)
boot

## End(Not run)
```

`inter_design` *Declare Survey Design Information and Census Margins*

Description

Validate the two inputs required by `fit_inter_sae()`. Column names are supplied as strings to keep programmatic workflows unambiguous.

Usage

```
inter_design(data, weights, area, strata = NULL, psu = NULL,
  replicate_weights = NULL)

census_margins(margins, count = "Freq", structural_zeros = NULL,
  tolerance = 1e-8, reconcile = FALSE)

## S3 method for class 'inter_design'
print(x, ...)

## S3 method for class 'census_margins'
print(x, ...)
```

Arguments

`data`	Survey microdata.
`weights`	Name of the survey-weight column.
`area`	Name of the geographic-area column.
`strata`	Optional stratum-column name.
`psu`	Optional primary-sampling-unit column name.
`replicate_weights`	Optional character vector naming replicate-weight columns.
`margins`	A census table or named list of census tables.
`count`	Common name of the population-count column.
`structural_zeros`	Optional data frame of impossible category combinations. Missing values act as wildcards.
`tolerance`	Relative tolerance for equality of census-table totals.
`reconcile`	Whether inconsistent table totals may be proportionally rescaled.
`x`	Object to print.
`...`	Unused.

Value

`inter_design()` returns an object of class `inter_design`. `census_margins()` returns an object of class `census_margins`.

Examples

```
example_data <- inter_example()
design <- inter_design(example_data$survey, weights = "weight",
  area = "area", strata = "strata", psu = "psu")
margins <- census_margins(example_data$margins)
design
margins
```

---

simulate_inter_data *Simulate an Intersectional Survey–Census Problem*

Description

Create a reproducible problem with a binary unemployment outcome and partially identified age-by-residence associations within area and sex groups.

Usage

```
simulate_inter_data(seed = 123, n = 500, areas = paste0("A", 1:6))

inter_example()
```

Arguments

`seed`	Random-number seed.
`n`	Survey sample size.
`areas`	Distinct area labels.

Value

A list containing survey microdata, overlapping census margins, and the latent full population table used to generate the data.

Examples

```
example_data <- inter_example()
names(example_data)
head(example_data$survey)
lapply(example_data$margins, head)
```

Index