

# Package ‘maSAE’

April 12, 2021

**Type** Package

**Title** Mandallaz' Model-Assisted Small Area Estimators

**Version** 2.0.3

**Description** An S4 implementation of the unbiased extension of the model- assisted synthetic-regression estimator proposed by Mandallaz (2013) <DOI:10.1139/cjfr-2012-0381>, Mandallaz et al. (2013) <DOI:10.1139/cjfr-2013-0181> and Mandallaz (2014) <DOI:10.1139/cjfr-2013-0449>. It yields smaller variances than the standard bias correction, the generalised regression estimator.

**License** BSD\_2\_clause + file LICENSE

**URL** <https://gitlab.com/fvafrcu/maSAE>

**Depends** R (>= 3.6.0)

**Imports** methods, stats

**Suggests** devtools, forestinventory, JoSAE, knitr, microbenchmark, nlme, pkgload, rmarkdown, rprojroot, rsae, RUnit, R.rsp, sae, testthat

**VignetteBuilder** utils, knitr, R.rsp

**RoxygenNote** 7.1.1

**NeedsCompilation** no

**Author** Andreas Dominik Cullmann [aut, cre],  
Daniel Mandallaz [ctb],  
Alexander Francis Massey [ctb]

**Maintainer** Andreas Dominik Cullmann <fvafrcu@mailbox.org>

**Repository** CRAN

**Date/Publication** 2021-04-12 09:40:02 UTC

## R topics documented:

maSAE-package . . . . .	2
predict . . . . .	3

s0	5
s1	5
s2	6
sadObj-class	7
saeObj-class	8
saObj	9

<b>Index</b>	<b>11</b>
--------------	-----------

---

maSAE-package	<i>Mandallaz' Model-Assisted Small Area Estimators</i>
---------------	--

---

## Description

An S4 implementation of the unbiased extension of the model-assisted' synthetic-regression estimator proposed by Mandallaz (2013), Mandallaz et al. (2013) and Mandallaz (2014). It yields smaller variances than the standard bias correction, the generalised regression estimator.

## Details

This package provides Mandallaz' extended synthetic-regression estimator for two- and three-phase sampling designs with or without clustering. See vignette("maSAE", package = "maSAE") and demo("maSAE", package = "maSAE") for introductions, "`class?maSAE::saeObj`" and "`?maSAE::predict`" for help on the main feature.

## Note

Model-assisted estimators use models to improve the efficiency (i.e. reduce prediction error compared to design-based estimators) but need not assume them to be correct as in the model-based approach, which is advantageous in official statistics.

## References

- Mandallaz, D. 2013 *Design-based properties of some small-area estimators in forest inventory with two-phase sampling*. *Canadian Journal of Forest Research* **43**(5), pp. 441–449. doi: [10.1139/cjfr20120381](https://doi.org/10.1139/cjfr20120381).
- Mandallaz, and Breschan, J. and Hill, A. 2013 *New regression estimators in forest inventories with two-phase sampling and partially exhaustive information: a design-based Monte Carlo approach with applications to small-area estimation*. *Canadian Journal of Forest Research* **43**(11), pp. 1023–1031. doi: [10.1139/cjfr20130181](https://doi.org/10.1139/cjfr20130181).
- Mandallaz, D. 2014 *A three-phase sampling extension of the generalized regression estimator with partially exhaustive information*. *Canadian Journal of Forest Research* **44**(4), pp. 383–388. doi: [10.1139/cjfr20130449](https://doi.org/10.1139/cjfr20130449).

**See Also**

There are a couple packages for model-based small area estimation, see [sae](#), [rsae](#), [hbsae](#) and [JoSAE](#). In 2016, Andreas Hill published [forestinventory](#), another implementation of Mandallaz' model-assisted small area estimators (see `vignette("forestinventory_and_maASE", package = "maSAE")` for a comparison).

**Examples**

```
## Not run:
vignette("maSAE", package = "maSAE")

## End(Not run)
## Not run:
demo("design", package = "maSAE")

## End(Not run)
## Not run:
demo("maSAE", package = "maSAE")

## End(Not run)
```

---

predict

*Methods for Function predict*


---

**Description**

Calculate small area predictions and their variances.

**Usage**

```
predict(object, ...)

## S4 method for signature 'sadObj'
predict(object)

## S4 method for signature 'saeObj'
predict(object, version = NULL, use_lm = NA)
```

**Arguments**

object	a model object for which prediction is desired.
...	Arguments to be passed to methods.
version	set to "1.0.0" or set <code>options(maSAE_version = "1.0.0")</code> to use the functions from maSAE 1.0.0. See NEWS.md for 2.0.0.
use_lm	Rather for internal use, stick with the default.

## Details

Based on the structure of the `saeObj` given, `predict` decides, which predictor to use:

If a `smallAreaMeans`-data.frame covering all fixed effects is given, the exhaustive estimator  $\hat{y}_{g, synth}$  is calculated.

If a `smallAreaMeans`-data.frame not covering all fixed effects is given, the partially exhaustive estimator  $\hat{y}_{g, greg}$  is calculated.

If no `smallAreaMeans`-data.frame but `s1` is given, the three-phase estimator  $\hat{y}_{g, g3reg}$  is calculated.

If neither `smallAreaMeans` nor `s1` are given, the non-exhaustive estimator  $\hat{y}_{g, psynth}$  is calculated.

If a clustering variable is given, the cluster sampling design equivalents of the above estimators are used.

If `version` is not set to "1.0.0", the (pseudo) small and synthetic estimations and their variances are also calculated (see `vignette("A_Taxonomy_of_Estimators", package = "maSAE")`)

## Value

A data frame containing predictions and variances for each small area, see **Details** above.

## Methods

`signature(object = saeObj)` Calculate predictions and variances according to the auxiliary information given, see **Details** above.

`signature(object = sadObj)` Calculate design-based predictions and variances.

## See Also

`vignette(package = "maSAE")`

## Examples

```
## ## design-based estimation
## load data
data("s2", package = "maSAE")
## create object
sae0 <- maSAE::saObj(data = s2, f = y ~ NULL | g)
## design-based estimation for all small areas given by g
maSAE::predict(sae0)
## ## model-assisted estimation
## load data
data("s1", "s2", package = "maSAE")
str(s1)
s12 <- maSAE::bind_data(s1, s2)
## create object
sae0 <- maSAE::saObj(data = s12, f = y ~ x1 + x2 + x3 | g, s2 = "phase2")
## small area estimation
maSAE::predict(sae0)
```

---

s0

*Example s0 Data Set*

---

**Description**

Artificial null phase sampling data used for examples in **maSAE**.

**Usage**

```
data(s0, package = "maSAE")
```

**Format**

A data frame with 9008 observations on the following 6 variables.

**Details**

clustid See "[?maSAE::s2](#)"

x1 See "[?maSAE::s2](#)"

x2 See "[?maSAE::s2](#)"

x3 See "[?maSAE::s2](#)"

inclusion See "[?maSAE::s2](#)"

g See "[?maSAE::s2](#)"

---

s1

*Example s1 Data Set*

---

**Description**

Artificial first phase sampling data used for examples in **maSAE**.

**Usage**

```
data(s1, package = "maSAE")
```

**Format**

A data frame with 786 observations on the following 6 variables.

**Details**

clustid See "?maSAE::s2"

x1 See "?maSAE::s2"

x2 See "?maSAE::s2"

x3 See "?maSAE::s2"

inclusion See "?maSAE::s2"

g See "?maSAE::s2"

---

s2

*Example s2 Data Set*

---

**Description**

Artificial second phase sampling data used for examples in **maSAE**.

**Usage**

```
data(s2, package = "maSAE")
```

**Format**

A data frame with 206 observations on the following 7 variables.

**Details**

clustid index giving the clusters.

x1 a potential fixed effect.

x2 another potential fixed effect.

x3 yet another potential fixed effect.

y the predictand

inclusion a logical vector indicating whether or not to include the current observation. All TRUE.

g A factor defining the small areas 'a' and 'b'

---

`sadObj-class`*Small Area Design-based Objects*

---

## Description

A class for design-based estimation only.

## Details

See "[saeObj](#)". The fixed effects part of `f` has to be `NULL`: design-based estimation knows no fixed effects.

## Slots

`data` See "[saeObj](#)".

`f` See "[saeObj](#)".

`cluster` See "[saeObj](#)".

`include` See "[saeObj](#)".

## Extends

Class "[savObj](#)", directly.

## Objects from the Class

Objects can be created by calls of the form `new("sadObj", ...)` or via the constructor function "[?maSAE::saObj](#)".

## Methods

[predict](#)

## Note

The slots are described in "[class?maSAE::saeObj](#)", since this is the main class of the package.

## See Also

["saeObj"](#) ["?maSAE::saObj"](#)

Other classes: [characterOrNULL-class](#), [saeObj-class](#), [savObj-class](#)

## Examples

```
showClass("sadObj")
```

saeObj-class

*Small Area Estimation Objects***Description**

Class for small area estimation, the one you're probably looking for.

**Details**

`cluster` optionally gives the name of a variable in slot data from which the cluster information for clustered sample designs is to be read. See Manadallaz 2013, p. 445 for Details.

`include` optionally gives the name of a variable in slot data from which the inclusion indicator for cluster points is to be read. See Manadallaz 2013, p. 445 for Details on  $I_f$ .

Also see the **Details** for `predict`.

**Slots**

`smallAreaMeans` An *optional* "data.frame" giving the true means of fixed effects for the small areas. Must have a column with the random effect defining the small areas in slot data.

`s1` An *optional* "character" string giving the name of a variable in slot data indicating that an observation (a row in slot data) belongs to subset 1.

`s2` An *optional* "character" string giving the name of a variable in slot data indicating that an observation (a row in slot data) belongs to subset 2.

`data` Object of class "data.frame" to use for prediction, typically consisting of a predictand and one or more predictors (zero or more fixed effects and one random effect defining the small areas). See **Details** for optional clustering variable and/or inclusion indicator.

`f` Object of class "formula" a linear mixed effects model formula.

`cluster` An *optional* "character" string giving the name of the clustering variable in slot data.

`include` An *optional* "character" string giving the name of the inclusion indicator in slot data.

`auxiliaryWeights` An *optional* "character" string giving the name of the auxiliary weights in slot data. You will need it, if your auxiliary data does not have full spatial support for each observation (for example when a shapefile does not completely cover all grid cells used to compute auxiliary data on). See `vignette("forestinventory_vignette", package = "forestinventory")` for details.

**Extends**

Class "`savObj`", directly.

**Objects from the Class**

Objects can be created by calls of the form `new("saeObj", ...)` or via the constructor function "`?maSAE::saObj`" (recommended).



**Methods**[predict](#)**References**

Mandallaz, D. 2013 *Design-based properties of some small-area estimators in forest inventory with two-phase sampling*. *Canadian Journal of Forest Research* **43**(5), pp. 441–449. doi: [10.1139/cjfr20120381](https://doi.org/10.1139/cjfr20120381).

**See Also**

["?stats::formula"](#), ["class?maSAE::saObj"](#), ["class?maSAE::savObj"](#), ["?maSAE::saObj"](#) and ["?maSAE::predict"](#)

Other classes: [characterOrNULL-class](#), [sadObj-class](#), [savObj-class](#)

**Examples**

```
showClass("saeObj")
```

---

saObj

*A Constructor for Objects of Class sadObj and saeObj*


---

**Description**

Simple wrapper to `new("sa[de]Obj")`. If missing, it adds an inclusion variable to data; it checks for missing in the clustering variable. Adds comments documenting changes made to the returned object.

**Usage**

```
saObj(
  data,
  f,
  smallAreaMeans = NULL,
  s1 = NULL,
  s2 = NULL,
  cluster = NULL,
  include = NULL,
  auxiliaryWeights = NULL
)
```

**Arguments**

data            See ["saeObj"](#).

f                a linear mixed effects formula, but see **Value**.

smallAreaMeans See ["saeObj"](#).

s1            See "saeObj".  
s2            See "saeObj".  
cluster       See "saeObj".  
include       See "saeObj".  
auxiliaryWeights    See "saeObj".

**Value**

An object of class sadObj if f is of structure 'x ~ NULL | g', an object of class saeObj otherwise.

**See Also**

"saeObj", "sadObj".

**Examples**

```
## load data
data("s2", package = "maSAE")
## create sadObj object
sad <- maSAE::saObj(data = s2, f = y ~ NULL | g)
class(sad)
## create saeObj object
s2$s2 <- TRUE
sae <- maSAE::saObj(data = s2, f = y ~ x1 + x2 + x3 | g, s2 = "s2")
class(sae)
```

# Index

- \* **classes**
  - sadObj-class, 7
  - saeObj-class, 8
- \* **datasets**
  - s0, 5
  - s1, 5
  - s2, 6
- \* **methods**
  - predict, 3
- \* **package**
  - maSAE-package, 2
  - ?maSAE::predict, 2, 9
  - ?maSAE::s2, 5, 6
  - ?maSAE::saObj, 7–9
  - ?stats::formula, 9
  
- class?maSAE::saeObj, 2, 7
- class?maSAE::saObj, 9
- class?maSAE::savObj, 9
  
- forestinventory, 3
  
- JoSAE, 3
  
- maSAE-package, 2
  
- predict, 3, 7–9
- predict, sadObj-method (predict), 3
- predict, saeObj-method (predict), 3
- predict-methods (predict), 3
  
- rsae, 3
  
- s0, 5
- s1, 5
- s2, 6
- sadObj, 10
- sadObj-class, 7
- sae, 3
- saeObj, 7, 9, 10
- saeObj-class, 8
  
- saObj, 9
- savObj, 7, 8