Package 'CalibrateSSB'

October 12, 2022

Type Package

Title Weighting and Estimation for Panel Data with Non-Response

Version 1.3.0

Date 2020-08-03

Author Øyvind Langsrud

Maintainer Oyvind Langsrud <oyl@ssb.no>

Depends R (>= 3.0.0), survey

Imports methods

Suggests ReGenesees, testthat (>= 2.1.0)

Description Functions to calculate weights, estimates of changes and corresponding variance estimates for panel data with non-response. Partially overlapping samples are handled. Initially, weights are calculated by linear calibration. By default, the survey package is used for this purpose. It is also possible to use ReGenesees, which can be installed from <https://github.com/DiegoZardetto/ReGenesees>. Variances of linear combinations (changes and averages) and ratios are calculated from a covariance matrix based on residuals according to the calibration model. The methodology was presented at the conference, The Use of R in Official Statistics, and is described in Langsrud (2016) <http://www.revistadestatistica.ro/wp-content/uploads/2016/06/RRS2_2016_A021.pdf>.

License GPL-2

RoxygenNote 7.1.1

Encoding UTF-8

URL https://github.com/statisticsnorway/CalibrateSSB

BugReports https://github.com/statisticsnorway/CalibrateSSB/issues

NeedsCompilation no

Repository CRAN

Date/Publication 2020-08-04 22:04:14 UTC

R topics documented:

`CalibrateSSB-package` *Weighting and Estimation for Panel Data with Non-Response*

Description

CalibrateSSB is an R-package that handles repeated surveys with partially overlapping samples. Initially the samples are weighted by linear calibration using known or estimated population totals. A robust model based covariance matrix for all relevant estimated totals is calculated from the residuals according to the calibration model. Alternatively a design based covariance matrix is calculated in a very similar way. A cluster robust version is also possible. In the case of estimated populations totals the covariance matrix is adjusted by utilizing the theory of Särndal and Lundström (2005). Variances of linear combinations (changes and averages) and ratios are calculated from this covariance matrix. The linear combinations and ratios can involve variables within and/or between sample waves.

References

Langsrud, Ø (2016): "A variance estimation R-package for repeated surveys - useful for estimates of changes in quarterly and annual averages", *Romanian Statistical Review* nr. 2 / 2016, pp. 17-28. CONFERENCE: *New Challenges for Statistical Software - The Use of R in Official Statistics*, Bucharest, Romania, 7-8 April.

Särndal, C.-E. and Lundström, S. (2005): *Estimation in Surveys with Nonresponse*, John Wiley and Sons, New York.

AkuData *Generate test data*

Description

Generate test data of eight quarters

Usage

```
AkuData(n)
```

Arguments

n	Number of observations within each quarter.

Value

A data frame with the following variables:

id	Sample unit identifier
year	Year
q	Quarter
month	Month
R	Response indicator
age	Age group
sex	Education group
famid	Family identifier
unemployed	Unemployed
workforce	In workforce

Examples

```
# Generates data - two years
z = AkuData(3000) # 3000 in each quarter
```

`CalibrateSSB` *Calibration weighting and estimation*

Description

Compute weights by calibration and corresponding estimates, totals and residuals

Usage

```
CalibrateSSB(
  grossSample,
  calmodel = NULL,
  response = "R",
  popTotals = NULL,
  y = NULL,
  by = NULL,
  partition = NULL,
  lRegmodel = NULL,
  popData = NULL,
  samplingWeights = NULL,
  usePackage = "survey",
  bounds = c(-Inf, Inf),
  calfun = "linear",
  onlyTotals = FALSE,
  onlyw = FALSE,
  uselRegWeights = FALSE,
  ids = NULL,
  residOutput = TRUE,
  leverageOutput = FALSE,
  yOutput = TRUE,
  samplingWeightsOutput = FALSE,
  dropResid2 = TRUE,
  wGrossOutput = TRUE,
  wave = NULL,
  id = NULL,
  extra = NULL,
  allowNApopTotals = NULL,
  partitionPrint = NULL,
  ...
)
```

Arguments

`grossSample`	Data frame.
`calmodel`	Formula defining the linear structure of the calibration model.
`response`	Variable name of response indicator (net sample when 1).

popTotals	Population totals (similar to population totals as output).
y	Names of variables of interest. Can be a list similar to "by" below.
by	Names of the variables that define the "estimation domains". If NULL (the default option) or NA estimates refer to the whole population. Use list for multiple specifications (resulting in list as output).
partition	Names of the variables that define the "calibration domains" for the model. NULL (the default) implies no calibration domains.
lRegmodel	Formula defining the linear structure of a logistic regression model.
popData	Data frame of population data.
samplingWeights	Name of the variable with initial weights for the sampling units.
usePackage	Specifying the package to be used: "survey" (the default), "ReGenesees" or "none".
bounds	Bounds for the calibration weights. When ReGenesees: Allowed range for the ratios between calibrated and initial weights. The default is c(-Inf,Inf).
calfun	The distance function for the calibration process; the default is 'linear'.
onlyTotals	When TRUE: Only population totals are returned.
onlyw	When TRUE: Only the calibrated weights are returned.
uselRegWeights	When TRUE: Weighted logistic regression is performed as a first calibration step.
ids	Name of sampling unit identifier variable.
residOutput	Residuals in output when TRUE. FALSE is default.
leverageOutput	Leverages in output when TRUE. FALSE is default.
yOutput	y in output when TRUE. FALSE is default.
samplingWeightsOutput	samplingWeights in output when TRUE. FALSE is default.
dropResid2	When TRUE (default) and when no missing population totals - only one set of residuals in output.
wGrossOutput	wGross in output when TRUE (default) and when NA popTotals.
wave	Time or another repeat variable (to be included in output).
id	Identifier variable (to be included in output).
extra	Variables for the extra dataset (to be included in output).
allowNApopTotals	When TRUE missing population totals are allowed. Results in error when FALSE and warning when NULL.
partitionPrint	When TRUE partition progress is printed. Automatic decision when NULL (about 1 min total computing time).
...	Further arguments sent to underlying functions.

Details

When popTotals as input is NULL, population totals are computed from popData (when available) or from grossSample. Some elements of popTotals may be missing (not allowed when using ReGenesees). When using "ReGenesees", both weiging and estimation are done by that package. When using "survey", only calibration weiging are done by that package. The parameters `wave`, `id` and `extra` have no effect on the computations, but result in extra elements in output (to be used by WideFromCalibrate() later).

Value

Unless onlyTotals or onlyw is TRUE, the output is an object of class calSSB. That is, a list with elements:

`popTotals`	Population totals.
`w`	The calibrated weights.
`wGross`	Calibrated gross sample weights when NA popTotals.
`estTM`	Estimates (with standard error).
`resids`	Residuals, reduced model when NA popTotals.
`resids2`	Residuals, full model.
`leverages`	Diagonal elements of hat-matrix, reduced model when NA popTotals.
`leverages2`	Diagonal elements of hat-matrix, full model.
`y`	as input
`samplingWeights`	as input
`wave`	as input or via CrossStrata
`id`	as input
`extra`	as input

See Also

`CalSSBobj`, `WideFromCalibrate`, `PanelEstimation`, `CalibrateSSBpanel`.

Examples

```
# Generates data  - two years
z    <- AkuData(3000)  # 3000 in each quarter
zPop <- AkuData(10000)[,1:7]

# Calibration using "survey"
a <- CalibrateSSB(z, calmodel = "~ sex*age",
                 partition = c("year","q"),  # calibrate within quarter
                 popData = zPop, y = c("unemployed","workforce"),
                 by = c("year","q")) # Estimate within quarter
head(a$w) # calibrated weights
a$estTM   # estimates
```

```
a$popTotals   # popTotals used as input below


# Calibration, no package, popTotals as input
b <- CalibrateSSB(z, popTotals=a$popTotals, calmodel="~ sex*age",
      partition = c("year","q"), usePackage = "none", y = c("unemployed","workforce"))
max(abs(a$w-b$w)) # Same weights as above

print(a)
print(b)

## Not run:
require(ReGenesees)
# Calibration and estimation via ReGenesees
CalibrateSSB(z, calmodel = "~ sex*age",
             partition = c("year","q"),  # calibrate within quarter
             popData = zPop, usePackage = "ReGenesees",
             y = c("unemployed","workforce"),
             by = c("year","q")) # Estimate within quarter

## End(Not run)
```

`CalibrateSSBpanel` *Calibration weighting and variance estimation for panel data*

Description

Calibration weighting and variance estimation for panel data

Usage

```
CalibrateSSBpanel(...)
```

Arguments

`...`	Input to CalibrateSSB() and PanelEstimation()

Value

Output from PanelEstimation()

See Also

`CalibrateSSB`, `PanelEstimation`.

Examples

```
z    = AkuData(3000)  # 3000 in each quarter
zPop = AkuData(10000)[,1:7]
lc = rbind(LagDiff(8,4),PeriodDiff(8,4))
rownames(lc) = c("diffQ1","diffQ2","diffQ3","diffQ4","diffYearMean")
CalibrateSSBpanel(grossSample=z,calmodel="~ sex*age", partition=c("year","q"),popData=zPop,
       y=c("unemployed","workforce"),id="id",wave=c("year","q"),
       numerator="unemployed",linComb=lc)
```

CalSSBobj *Create or modify a CalSSB object*

Description

The elements of the CalSSB object are taken directly from the input parameters.

Usage

```
CalSSBobj(
  x = NULL,
  y = NULL,
  w = NULL,
  wGross = NULL,
  resids = NULL,
  resids2 = NULL,
  leverages = NULL,
  leverages2 = NULL,
  samplingWeights = NULL,
  extra = NULL,
  id = NULL,
  wave = NULL
)
```

Arguments

x	NULL or an existing calSSB object
y	y
w	w
wGross	wGross
resids	resids
resids2	resids2
leverages	leverages
leverages2	leverages2
samplingWeights	samplingWeights

extra	extra
id	id
wave	wave

Value

A CalSSB object. That is, an object of the type retuned by CalibrateSSB.

Note

If x is a ReGenesees/cal.analytic object, this function is a wrapper to CalSSBobjReGenesees.

See Also

CalibrateSSB, CalSSBobjReGenesees, WideFromCalibrate, PanelEstimation.

Examples

```
#' # Generates data - two years
z <- AkuData(3000)  # 3000 in each quarter
zPop <- AkuData(10000)[, 1:7]

# Create a CalSSB object by CalibrateSSB
b <- CalibrateSSB(z, calmodel = "~ sex*age", partition = c("year", "q"), popData = zPop,
                  y = c("unemployed", "workforce"))

# Modify the CalSSB object
a <- CalSSBobj(b, w = 10*b$w, wave = CrossStrata(z[, c("year", "q")]), id = z$id)

# Use the CalSSB object as input ...
PanelEstimation(WideFromCalibrate(a), "unemployed", linComb = PeriodDiff(8, 4))

# Create CalSSB object without x as input
CalSSBobj(y = b$y, w = 10*b$w, resids = b$resids, wave = CrossStrata(z[, c("year", "q")]),
          id = z$id)
```

CalSSBobjReGenesees *Create a CalSSB object from a ReGenesees/cal.analytic object*

Description

Create a CalSSB object from a ReGenesees/cal.analytic object

Usage

```
CalSSBobjReGenesees(
  x,
  y,
  samplingWeights = NULL,
  extra = NULL,
  id = NULL,
  wave = NULL
)
```

Arguments

x	Output from ReGenesees::e.calibrate() (object of class cal.analytic)
y	formula or variable names
samplingWeights	NULL, TRUE (capture from x), formula, variable name or vector of data
extra	NULL, formula, variable names or matrix of data
id	NULL, TRUE (ids from x), formula, variable name or vector of data
wave	NULL, formula, variable name or vector of data

Value

A CalSSB object. That is, an object of the type retuned by CalibrateSSB.

See Also

CalibrateSSB, CalSSBobj, WideFromCalibrate, PanelEstimation.

Examples

```
## Not run:
# Generates data - two years
z <- AkuData(3000)  # 3000 in each quarter
zPop <- AkuData(10000)[, 1:7]
z$samplingWeights <- 1
z$ids <- 1:NROW(z)

# Create a ReGenesees/cal.analytic object
library("ReGenesees")
desReGenesees <- e.svydesign(z[z$R == 1, ], ids = ~ids, weights = ~samplingWeights)
popTemplate <- pop.template(data = desReGenesees, calmodel = ~sex * age, partition = ~year + q)
popTotals <- fill.template(universe = zPop, template = popTemplate)
calReGenesees <- e.calibrate(design = desReGenesees, df.population = popTotals)

# Create CalSSB objects from a ReGenesees/cal.analytic object
CalSSBobjReGenesees(calReGenesees, y = ~unemployed + workforce, id = TRUE,
                    samplingWeights = TRUE, extra = ~famid)
a <- CalSSBobjReGenesees(calReGenesees, y = c("unemployed", "workforce"),
                         id = "id", extra = "famid", wave = c("year", "q"))
```

```
# Use the CalSSB object as input ...
PanelEstimation(WideFromCalibrate(a), "unemployed", linComb = PeriodDiff(8, 4))


## End(Not run)
```

CrossStrata *Crossing several factor variables*

Description

Create new factor variable by crossing levels in several variables

Usage

```
CrossStrata(by, sep = "-", returnb = FALSE, asNumeric = FALSE, byExtra = NULL)
```

Arguments

by	Dataframe or matrix with several variables
sep	Used to create new level names
returnb	When TRUE an overview of original variabels according to new levels are also retuned.
asNumeric	When TRUE the new variable is numeric.
byExtra	Contains the same variables as by and represents another data set.

Value

a	The new variable
aExtra	New variable according to byExtra
b	Overview of original variabels according to new levels

Examples

```
CrossStrata(cbind(factor(rep(1:3,2)),c('A',rep('B',5)) ))
```

`LinCombMatrix` *Creation of linear combination matrices*

Description

Create matrices for changes (LagDiff), means (Period) and mean changes (PeriodDiff).

Usage

```
LinCombMatrix(
  n,
  period = NULL,
  lag = NULL,
  k = 0,
  takeMean = TRUE,
  removerows = TRUE,
  overlap = FALSE
)

LagDiff(n, lag = 1, removerows = TRUE)

Period(
  n,
  period = 1,
  k = 0,
  takeMean = TRUE,
  removerows = TRUE,
  overlap = FALSE
)

PeriodDiff(
  n,
  period = 1,
  lag = period,
  k = 0,
  takeMean = TRUE,
  removerows = TRUE,
  overlap = FALSE
)
```

Arguments

`n`	Number of variables
`period`	Number of variables involved in each period
`lag`	Lag used for difference calculation
`k`	Shift the start of each period

takeMean	Calculate mean over each period (sum when FALSE)
removerows	Revove incomplete rows
overlap	Overlap between periods (moving averages)

Value

Linear combination matrix

Note

It can be useful to add row names to the resulting matrix before further use.

Examples

```
# We assume two years of four quarters (n=8)

# Quarter to quarter differences
LagDiff(8)

# Changes from same quarter last year
LagDiff(8,4)

# Yearly averages
Period(8,4)

# Moving yearly averages
Period(8,4,overlap=TRUE)

# Difference between yearly averages
PeriodDiff(8,4) # Also try n=16 with overlap=TRUE/FALSE

# Combine two variants and add row names
lc = rbind(LagDiff(8,4),PeriodDiff(8,4))
rownames(lc) = c("diffQ1","diffQ2","diffQ3","diffQ4","diffYearMean")
lc
```

PanelEstimation *Variance estimation for panel data*

Description

Variance estimation of linear combinations of totals and ratios based on output from wideFromCalibrate

Usage

```
PanelEstimation(
  x,
  numerator,
  denominator = NULL,
  linComb = matrix(0, 0, n),
  linComb0 = NULL,
  estType = "robustModel",
  leveragePower = 1/2,
  group = NULL,
  returnCov = FALSE,
  usewGross = TRUE
)
```

Arguments

x	Output from wideFromCalibrate.
numerator	y variable name or number.
denominator	y variable name or number.
linComb	Matrix defining linear combinations of waves.
linComb0	Linear combination matrix to be used prior to ratio calculations.
estType	Estimation type: "robustModel" (default), "ssbAKU", "robustModelww", "robustModelGroup" or "robustModelGroupww" (see below)
leveragePower	Power used when adjusting residuals using leverages.
group	Extra variable name or number for cluster robust estimation.
returnCov	Return covariance matrices instead of variance vectors.
usewGross	Use wGross (if avaliable) instead of design weights to adjust covariance matrix in the case of NA popTotals

Details

When denominator=NULL, only estimates for a single y-variable (numerator) are calculated. When denominator is specified, estimates for numerator, denominator and ratio are calculated. The default estimation type parameter, "robustModel", is equation (12) in paper. "ssbAKU" is (16), "robustModelww" is (9) and "robustModelGroup" and "robustModelGroupww" are cluster robust variants based on $(w-1)^2$ and w^2 .

Value

wTot	Sum of weights
estimates	Ordinary estimates
linCombs	Estimates of linear combinations
varEstimates	Variance of estimates
varLinCombs	Variance of estimates of linear combinations

When denominator is specified the above output refer to ratios. Then, similar output for numerator and denominator are also included.

See Also

CalibrateSSB, CalSSBobj, WideFromCalibrate, CalibrateSSBpanel.

Examples

```
# Generates data  - two years
z    = AkuData(3000)  # 3000 in each quarter
zPop = AkuData(10000)[,1:7]

# Calibration and "WideFromCalibrate"
b = CalibrateSSB(z,calmodel="~ sex*age", partition=c("year","q"),
        popData=zPop, y=c("unemployed","workforce"))
bWide = WideFromCalibrate(b,CrossStrata(z[,c("year","q")]),z$id)

# Define linear combination matrix
lc = rbind(LagDiff(8,4),PeriodDiff(8,4))
rownames(lc) = c("diffQ1","diffQ2","diffQ3","diffQ4","diffYearMean")
colnames(lc) = colnames(head(bWide$y[[1]]))
lc

# Unemployed: Totals and linear combinations
d1=PanelEstimation(bWide,"unemployed",linComb=lc)  #

# Table of output
cbind(tot=d1$estimates,se=sqrt(d1$varEstimates))
cbind(tot=d1$linCombs,se=sqrt(d1$varLinCombs))

# Ratio: Totals and linear combinations
d=PanelEstimation(bWide,numerator="unemployed",denominator="workforce",linComb=lc)
cbind(tot=d$estimates,se=sqrt(d$varEstimates))
cbind(tot=d$linCombs,se=sqrt(d$varLinCombs))

## Not run:
# Calibration when som population totals unknown (edu)
# Leverages in output (will be used to adjust residuals)
# Cluster robust estimation (families/famid)
b2 = CalibrateSSB(z,popData=zPop,calmodel="~ edu*sex + sex*age",
           partition=c("year","q"), y=c("unemployed","workforce"),
           leverageOutput=TRUE)
b2Wide = WideFromCalibrate(b2,CrossStrata(z[,c("year","q")]),z$id,extra=z$famid)
d2 = PanelEstimation(b2Wide,"unemployed",linComb=lc,group=1,estType = "robustModelGroup")
cbind(tot=d2$linCombs,se=sqrt(d2$varLinCombs))

## End(Not run)


# Yearly mean before ratio calculation (linComb0)
# and difference between years (linComb)
g=PanelEstimation(bWide,numerator="unemployed",denominator="workforce",
    linComb= LagDiff(2),linComb0=Period(8,4))
cbind(tot=g$linCombs,se=sqrt(g$varLinCombs))
```

print.calSSB *Print method for calSSB*

Description

Print method for calSSB

Usage

```
## S3 method for class 'calSSB'
print(x, digits = max(getOption("digits") - 3, 3), ...)
```

Arguments

x	calSSB object
digits	positive integer. Minimum number of significant digits to be used for printing most numbers.
...	further arguments sent to the underlying

Value

Invisibly returns the original object.

print.calSSBwide *Print method for calSSBwide*

Description

Print method for calSSBwide

Usage

```
## S3 method for class 'calSSBwide'
print(x, digits = max(getOption("digits") - 3, 3), ...)
```

Arguments

x	calSSBwide object
digits	positive integer. Minimum number of significant digits to be used for printing most numbers.
...	further arguments sent to the underlying

Value

Invisibly returns the original object.

`WideFromCalibrate` *Rearrange output from CalibrateSSB (calSSB object). Ready for input to PanelEstimation.*

Description

One row for each id and one column for each wave.

Usage

```
WideFromCalibrate(a, wave = NULL, id = NULL, subSet = NULL, extra = NULL)
```

Arguments

`a`	A calSSB object. That is, output from CalibrateSSB() or CalSSBobj().
`wave`	Time or another repeat variable.
`id`	Identifier variable.
`subSet`	Grouping variable for splitting ouput.
`extra`	Dataset with extra variables not in `a`.

Details

When wave, id or extra is NULL, corresponding elements in the input object (`a`) will be used if available,

Value

Output has the same elements (+ extra) as input (a), but rearranged. When subSet is input otput is alist according to the subSet levels.

See Also

`CalibrateSSB`, `CalSSBobj`, `PanelEstimation`.

Examples

```
# See examples in PanelEstimation and CalSSBobj
```

Index