Package 'sae'

November 4, 2026

Type Package

Title Small Area Estimation

Version 1.4

Date 2026-11-03

Depends stats, MASS, lme4

Description Functions for small area estimation.

License GPL-2

NeedsCompilation no

Repository CRAN

Date/Publication 2026-11-04 03:20:02 UTC

Author Isabel Molina [aut],
Yolanda Marhuenda [aut, cre]

Maintainer Yolanda Marhuenda <y.marhuenda@umh.es>

Contents

sae-package 2
bxcx 4
cornsoybean 5
cornsoybeanmeans 6
diagonalizematrix 6
direct 7
ebBHF 9
eblupBHF 11
eblupFH 13
eblupSFH 15
eblupSTFH 17
grapes 20
grapesprox 21
incomedata 21
milk 22
mseFH 23

mseSFH 25
npbmseSFH 26
pbmseBHF 28
pbmseebBHF 30
pbmseSFH 32
pbmseSTFH 34
pssynt 37
sizeprov 39
sizeprovage 39
sizeprovedu 40
sizeprovlab 40
sizeprovnat 41
spacetime 41
spacetimeprox 42
ssd 42
Xoutsamp 44

Index **45**

`sae-package` *Small area estimation*

Description

This package provides a variety of functions for small area estimation, including functions for mean squared error estimation. Basic estimators include direct, poststratified synthetic and sample size dependent. Model-based estimators include the EBLUP based on a Fay-Herriot model and the EBLUP based on a unit level nested error model. Estimators obtained from spatial and spatio-temporal Fay-Herriot models and the EB method based on the unit level nested error model for estimation of general non linear parameters are also included.

Details

This package provides functions for estimation in domains with small sample sizes. For a complete list of functions, see library(help=sae).

Package:	sae
Type:	Package
Version:	1.2
Date:	2018-05-01
License:	GPL-2
Depends:	stats, lmer

Author(s)

Isabel Molina <isabel.molina@uc3m.es> and Yolanda Marhuenda <y.marhuenda@umh.es>

References

- Arora, V. and Lahiri, P. (1997). On the superiority of the Bayesian method over the BLUP in small area estimation problems. Statistica Sinica 7, 1053-1063.

- Battesse, G.E., Harter, R.M. and Fuller, W.A. (1988). An Error-Components Model for Prediction of County Crop Areas Using Survey and Satellite Data. Journal of the American Statistical Association 83, 28-36.

- Box, G.E.P. and Cox, D.R. (1964). An analysis of transformations. Journal of Royal Statistical Society Series B 26, 211-246.

- Cochran, W.G. (1977). Sampling techniques. Wiley, New York.

- Datta, G.S. and Lahiri, P. (2000). A unified measure of uncertainty of estimated best linear unbiased predictors in small area estimation problems. Statistica Sinica 10, 613-627.

- Datta, G.S., Rao, J.N.K. and Smith D.D. (2005). On measuring the variability of small area estimators under a basic area level model. Biometrika 92, 183-196.

- Drew, D., Singh, M.P. and Choudhry, G.H. (1982). Evaluation of small area estimation techniques for the Canadian Labour Force Survey. Survey Methodology 8, 17-47.

- Fay, R.E. and Herriot, R.A. (1979). Estimation of income from small places: An application of James-Stein procedures to census data. Journal of the American Statistical Association 74, 269-277.

- Gonzalez-Manteiga, W., Lombardia, M., Molina, I., Morales, D. and Santamaria, L. (2008). Analytic and bootstrap approximations of prediction errors under a multivariate Fay-Herriot model. Computational Statistics and Data Analysis 52, 5242-5252.

- Jiang, J. (1996). REML estimation: asymptotic behavior and related topics. Annals of Statistics 24, 255-286.

- Marhuenda, Y., Molina, I. and Morales, D. (2013). Small area estimation with spatio-temporal Fay-Herriot models. Computational Statistics and Data Analysis 58, 308-325.

- Marhuenda, Y., Morales, D. and Pardo, M.C. (2014). Information criteria for Fay-Herriot model selection. Computational Statistics and Data Analysis 70, 268-280.

- Molina, I., Salvati, N. and Pratesi, M. (2009). Bootstrap for estimating the MSE of the Spatial EBLUP. Computational Statistics 24, 441-458.

- Molina, I. and Rao, J.N.K. (2010). Small Area Estimation of Poverty Indicators. The Canadian Journal of Statistics 38, 369-385.

- Petrucci, A. and Salvati, N. (2006). Small area estimation for spatial correlation in watershed erosion assessment. Journal of Agricultural, Biological and Environmental Statistics 11, 169-182.

- Prasad, N. and Rao, J. (1990). The estimation of the mean squared error of small-area estimators. Journal of the American Statistical Association 85, 163-171.

- Pratesi, M. and Salvati, N. (2008). Small area estimation: the EBLUP estimator based on spatially correlated random area effects. Statistical Methods & Applications 17, 113-141.

- Rao, J.N.K. (2003). Small Area Estimation. Wiley, London.

- Sarndal, C.E., Swensson, B. and Wretman, J. (1992). Model Assisted Survey Sampling. Springer-Verlag.

- Singh, B., Shukla, G. and Kundu, D. (2005). Spatio-temporal models in small area estimation. Survey Methodology 31, 183-195.

- Small Area Methods for Poverty and Living Conditions Estimates (SAMPLE), funded by European Commission, Collaborative Project 217565, Call identifier FP7-SSH-2007-1.

- You, Y. and Chapman, B. (2006). Small area estimation using area level models and estimated sampling variances. Survey Methodology 32, 97-103.

`bxcx` *Box-Cox Transformation and its Inverse*

Description

Box-Cox or power transformation or its inverse. For lambda!=0, the Box-Cox transformation of x is (x^lambda-1)/lambda, whereas the regular power transformation is simply x^lambda. When lambda=0, it is log in both cases. The inverse of the Box-Cox and the power transform can also be obtained.

Usage

```
bxcx(x, lambda, InverseQ = FALSE, type = "BoxCox")
```

Arguments

`x`	a vector or time series
`lambda`	power transformation parameter
`InverseQ`	if TRUE, the inverse transformation is done
`type`	either "BoxCox" or "power"

Value

A vector or time series of the transformed data

Author(s)

A.I. McLeod. R package `FitAR`

References

- Box, G.E.P. and Cox, D.R. (1964). An analysis of transformations. Journal of Royal Statistical Society Series B 26, 211-246.

Examples

```
#lambda=0.5
z<-AirPassengers; lambda<-0.5
y<-bxcx(z, lambda)
z2<-bxcx(y, lambda, InverseQ=TRUE)
sum(abs(z2-z))
#
z<-AirPassengers; lambda<-0.0
y<-bxcx(z, lambda)
z2<-bxcx(y, lambda, InverseQ=TRUE)
sum(abs(z2-z))
```

---

cornsoybean *Corn and soy beans survey and satellite data in 12 counties in Iowa.*

Description

Survey and satellite data for corn and soy beans in 12 Iowa counties, obtained from the 1978 June Enumerative Survey of the U.S. Department of Agriculture and from land observatory satellites (LANDSAT) during the 1978 growing season.

Usage

```
data(cornsoybean)
```

Format

A data frame with 37 observations on the following 5 variables.

`County`: numeric county code.

`CornHec`: reported hectares of corn from the survey.

`SoyBeansHec`: reported hectares of soy beans from the survey.

`CornPix`: number of pixels of corn in sample segment within county, from satellite data.

`SoyBeansPix`: number of pixels of soy beans in sample segment within county, from satellite data.

Source

- Battesse, G.E., Harter, R.M. and Fuller, W.A. (1988). An Error-Components Model for Prediction of County Crop Areas Using Survey and Satellite Data. Journal of the American Statistical Association 83, 28-36.

cornsoybeanmeans *Corn and soy beans mean number of pixels per segment for 12 counties in Iowa.*

Description

County means of number of pixels per segment of corn and soy beans, from satellite data, for 12 counties in Iowa. Population size, sample size and means of auxiliary variables in data set cornsoybean.

Usage

```
data(cornsoybeanmeans)
```

Format

A data frame with 12 observations on the following 6 variables.

`CountyIndex`: numeric county code.

`CountyName`: name of the county.

`SampSegments`: number of sample segments in the county (sample size).

`PopnSegments`: number of population segments in the county (population size).

`MeanCornPixPerSeg`: mean number of corn pixels per segment in the county.

`MeanSoyBeansPixPerSeg`: mean number of soy beans pixels per segment in the county.

Source

- Battesse, G.E., Harter, R.M. and Fuller, W.A. (1988). An Error-Components Model for Prediction of County Crop Areas Using Survey and Satellite Data. Journal of the American Statistical Association 83, 28-36.

diagonalizematrix *It constructs a block-diagonal matrix.*

Description

Using a `n*m` matrix `A`, this function constructs a block-diagonal matrix with dimension `(n*ntimes) * (m*ntimes)`, with all blocks equal to matrix `A` and the rest of entries equal to `0`.

Usage

```
diagonalizematrix(A, ntimes)
```

Arguments

A	n*m matrix with the values.
ntimes	number of times.

Examples

```
X <- matrix(data=c(1,2,3,4,5,6), nrow=3, ncol=2)
diagonalizematrix(X,3)
```

---

direct *Direct estimators.*

Description

This function calculates direct estimators of domain means.

Usage

```
direct(y, dom, sweight, domsize, data, replace = FALSE)
```

Arguments

y	vector specifying the individual values of the variable for which we want to estimate the domain means.
dom	vector or factor (same size as y) with domain codes.
sweight	optional vector (same size as y) with sampling weights. When this argument is not included, by default estimators are obtained under simple random sampling (SRS).
domsize	D*2 data frame with domain codes in the first column and the corresponding domain population sizes in the second column. This argument is not required when sweight is not included and replace=TRUE (SRS with replacement).
data	optional data frame containing the variables named in y, dom and sweight. By default the variables are taken from the environment from which direct is called.
replace	logical variable with default value FALSE for random sampling without replacement within each domain is considered and TRUE for random sampling with replacement within each domain.

Value

The function returns a data frame of size D*5 with the following columns:

Domain	domain codes in ascending order.
SampSize	domain sample sizes.
Direct	direct estimators of domain means of variable y.

SD — estimated standard deviations of domain direct estimators. If sampling design is SRS or Poisson sampling, estimated variances are unbiased. Otherwise, estimated variances are obtained under the approximation that second order inclusion probabilities are the product of first order inclusion probabilities.

CV — absolute value of percent coefficients of variation of domain direct estimators.

Cases with NA values in y, dom or sweight are ignored.

References

- Cochran, W.G. (1977). Sampling techniques. Wiley, New York.

- Rao, J.N.K. (2003). Small Area Estimation. Wiley, London.

- Sarndal, C.E., Swensson, B. and Wretman, J. (1992). Model Assisted Survey Sampling. Springer-Verlag.

See Also

pssynt for post-stratified synthetic estimator, ssd for sample size dependent estimator.

In case that the sampling design is known, see packages survey or sampling for more exact variance estimation.

Examples

```
# Load data set with synthetic income data for provinces (domains)
data(incomedata)

# Load population sizes of provinces
data(sizeprov)

# Compute Horvitz-Thompson direct estimator of mean income for each
# province under random sampling without replacement within each province.
result1 <- direct(y=income, dom=prov, sweight=weight,
                  domsize=sizeprov[,2:3], data=incomedata)
result1

# The same but using province labels as domain codes
result2 <- direct(y=incomedata$income, dom=incomedata$provlab,
                  sweight=incomedata$weight, domsize=sizeprov[,c(1,3)])
result2

# The same, under SRS without replacement within each province.
result3 <- direct(y=income ,dom=provlab, domsize=sizeprov[,c(1,3)],
                  data=incomedata)
result3

# Compute direct estimator of mean income for each province
# under SRS with replacement within each province
result4 <- direct(y=income, dom=provlab, data=incomedata, replace=TRUE)
result4
```

ebBHF *EB estimators of an indicator with non-sample values of auxiliary variables.*

Description

Fits by REML method the unit level model of Battese, Harter and Fuller (1988) to a transformation of the specified dependent variable by a Box-Cox family or power family and obtains Monte Carlo approximations of EB estimators of the specified small area indicators, when the values of auxiliary variables for out-of-sample units are available.

Usage

```
ebBHF(formula, dom, selectdom, Xnonsample, MC = 100, data,
      transform = "BoxCox", lambda = 0, constant = 0, indicator)
```

Arguments

formula	an object of class formula (or one that can be coerced to that class): a symbolic description of the model to be fitted. The details of model specification are given under Details.
dom	n*1 vector or factor (same size as y in formula) with domain codes.
selectdom	I*1 optional vector or factor with the domain codes for which we want to estimate the indicators. It must be a subset of the domain codes in dom. If this parameter is not included, the unique domain codes included in dom are considered.
Xnonsample	matrix or data frame containing in the first column the domain codes and in the rest of columns the values of each of p auxiliary variables for the out-of-sample units in each selected domain. The domains considered in Xnonsample must contain at least those specified in selectdom.
MC	number of Monte Carlo replicates for the empirical approximation of the EB estimator. Default value is MC=100.
data	optional data frame containing the variables named in formula and dom. By default the variables are taken from the environment from which ebBHF is called.
transform	type of transformation for the dependent variable to be chosen between the "BoxCox" and "power" families so that the dependent variable in formula follows approximately a Normal distribution. Default value is "BoxCox".
lambda	value for the parameter of the family of transformations specified in transform. Default value is 0, which gives the log transformation for the two possible families.
constant	constant added to the dependent variable before doing the transformation, to achieve a distribution close to Normal. Default value is 0.
indicator	function of the (untransformed) variable on the left hand side of formula that we want to estimate in each domain.

Details

This function uses random number generation. To fix the seed, use `set.seed`.

A typical model has the form response ~ terms where response is the (numeric) response vector and terms is a series of terms which specifies a linear predictor for response. A terms specification of the form first + second indicates all the terms in first together with all the terms in second with duplicates removed.

A formula has an implied intercept term. To remove this use either y ~ x - 1 or y ~ 0 + x. See `formula` for more details of allowed formulae.

Value

The function returns a list with the following objects:

`eb` data frame with number of rows equal to number of selected domains, containing in its columns the domain codes (`domain`), the EB estimators of `indicator` (`eb`) and the sample sizes (`sampsize`). For domains with zero sample size, the EB estimators are based on the synthetic regression. For domains in `selectdom` not included in `Xnonsample` the EB estimators are NA.

`fit` a list containing the following objects:

- `summary` summary of the unit level model fitting.
- `fixed` vector with the estimated values of the fixed regression coefficient.
- `random` vector with the predicted random effects.
- `errorvar` estimated model error variance.
- `refvar` estimated random effects variance.
- `loglike` log-likelihood.
- `residuals` vector with raw residuals from the model fit.

Cases with NA values in `formula` or `dom` are ignored.

References

- Molina, I. and Rao, J.N.K. (2010). Small Area Estimation of Poverty Indicators. The Canadian Journal of Statistics 38, 369-385.

See Also

`pbmseebBHF`

Examples

```
data(incomedata)         # Load data set
attach(incomedata)

# Construct design matrix for sample elements
Xs <- cbind(age2, age3, age4, age5, nat1, educ1, educ3, labor1, labor2)

# Select the domains to compute EB estimators.
data(Xoutsamp)
```

```
domains <- unique(Xoutsamp[,"domain"])

# Poverty gap indicator
povertyline <- 0.6*median(income)
povertyline                          # 6477.484
povgap <- function(y)
{
   z <- 6477.484
   result <- mean((y<z) * (z-y) / z)
   return (result)
}

# Compute EB predictors of poverty gap. The value constant=3600 is selected
# to achieve approximately symmetric residuals.
set.seed(123)
result <- ebBHF(income ~ Xs, dom=prov, selectdom=domains,
                Xnonsample=Xoutsamp, MC=10, constant=3600, indicator=povgap)
result$eb
result$fit$summary
result$fit$fixed
result$fit$random[,1]
result$fit$errorvar
result$fit$refvar
result$fit$loglike
result$fit$residuals[1:10]

detach(incomedata)
```

`eblupBHF` *EBLUPs of domain means based on a nested error linear regression model.*

Description

This function calculates, for selected domains, EBLUPs of domain means based on the nested error linear regression model of Battese, Harter and Fuller (1988).

Usage

```
eblupBHF(formula, dom, selectdom, meanxpop, popnsize, method = "REML", data)
```

Arguments

`formula`	an object of class `formula` (or one that can be coerced to that class): a symbolic description of the model to be fitted. The details of model specification are given under Details.
`dom`	`n*1` vector or factor (same size as `y` in `formula`) with domain codes.
`selectdom`	`I*1` optional vector or factor with the domain codes for which we want to estimate the means. It must be a subset of the domain codes in `dom`. If this parameter is not included all the domain codes included in `dom` are considered.

meanxpop	D*(p+1) data frame with domain codes in the first column. Each remaining column contains the population means of each of the p auxiliary variables for the D domains. The domains considered in meanxpop must contain those specified in selectdom (D>=I).
popnsize	D*2 data frame with domain codes in the first column and the corresponding domain population sizes in the second column. The domains considered in popnsize must contain those specified in selectdom (D>=I).
method	a character string. If "REML", the model is fitted by maximizing the restricted log-likelihood. If "ML" the log-likelihood is maximized. Defaults to "REML".
data	optional data frame containing the variables named in formula and dom. By default the variables are taken from the environment from which eblupBHF is called.

Details

A typical model has the form response ~ terms where response is the (numeric) response vector and terms is a series of terms which specifies a linear predictor for response. A terms specification of the form first + second indicates all the terms in first together with all the terms in second with duplicates removed.

A formula has an implied intercept term. To remove this use either y ~ x - 1 or y ~ 0 + x. See formula for more details of allowed formulae.

Value

The function returns a list with the following objects:

eblup	data frame with number of rows equal to number of selected domains (selectdom), containing in its columns the domain codes (domain) and the EBLUPs of the means of selected domains based on the nested error linear regression model (eblup). For domains with zero sample size, the EBLUPs are the synthetic regression estimators.
fit	a list containing the following objects: summary summary of the unit level model fitting. fixed vector with the estimated values of the fixed regression coefficient. random vector with the predicted random effects. errorvar estimated model error variance. refvar estimated random effects variance. loglike log-likelihood. residuals vector with raw residuals.

Cases with NA values in formula or dom are ignored.

References

- Battese, G.E., Harter, R.M. and Fuller, W.A. (1988). An Error-Components Model for Prediction of County Crop Areas Using Survey and Satellite Data, Journal of the American Statistical Association 83, 28-36.

- Rao, J.N.K. (2003). Small Area Estimation. New York: John Wiley and Sons.

See Also

pbmseBHF

Examples

```
# Load data set for segments (units within domains)
data(cornsoybean)

# Load data set for counties
data(cornsoybeanmeans)
attach(cornsoybeanmeans)

# Construct data frame with county means of auxiliary variables for
# domains. First column must include the county code
Xmean <- data.frame(CountyIndex, MeanCornPixPerSeg, MeanSoyBeansPixPerSeg)
Popn  <- data.frame(CountyIndex, PopnSegments)

# Compute EBLUPs of county means of corn crop areas for all counties
resultCorn <- eblupBHF(CornHec ~ CornPix + SoyBeansPix, dom=County,
                       meanxpop=Xmean, popnsize=Popn, data=cornsoybean)
resultCorn$eblup

# Compute EBLUPs of county means of soy beans crop areas for
# a subset of counties using ML method
domains <- c(10,1,5)
resultBean <- eblupBHF(SoyBeansHec ~ CornPix + SoyBeansPix, dom=County,
                       selectdom=domains, meanxpop=Xmean, popnsize=Popn,
                       method="ML", data=cornsoybean)
resultBean$eblup
resultBean$fit

detach(cornsoybeanmeans)
```

---

eblupFH *EBLUPs based on a Fay-Herriot model.*

Description

This function gives the EBLUP (or EB predictor under normality) based on a Fay-Herriot model. Fitting method can be chosen between ML, REML and FH methods.

Usage

```
eblupFH(formula, vardir, method = "REML", MAXITER = 100, PRECISION = 0.0001,
        B = 0, data)
```

Arguments

formula	an object of class formula (or one that can be coerced to that class): a symbolic description of the model to be fitted. The variables included in formula must have a length equal to the number of domains D. Details of model specification are given under Details.
vardir	vector containing the D sampling variances of direct estimators for each domain. The values must be sorted as the variables in formula.
method	type of fitting method, to be chosen between "ML", "REML" or "FH" methods.
MAXITER	maximum number of iterations allowed in the Fisher-scoring algorithm. Default is 100 iterations.
PRECISION	convergence tolerance limit for the Fisher-scoring algorithm. Default value is 0.0001.
B	number of bootstrap replicates to calculate the goodness-of-fit measures proposed by Marhuenda et al. (2014). Default value is 0 indicating that these measures are not calculated.
data	optional data frame containing the variables named in formula and vardir. By default the variables are taken from the environment from which eblupFH is called.

Details

A typical model has the form response ~ terms where response is the (numeric) response vector and terms is a series of terms which specifies a linear predictor for response. A terms specification of the form first + second indicates all the terms in first together with all the terms in second with duplicates removed.

A formula has an implied intercept term. To remove this use either y ~ x - 1 or y ~ 0 + x. See formula for more details of allowed formulae.

Value

The function returns a list with the following objects:

eblup	vector with the values of the estimators for the domains.
fit	a list containing the following objects:

method type of fitting method applied ("REML", "ML"or "FH").

convergence a logical value equal to TRUE if Fisher-scoring algorithm converges in less than MAXITER iterations.

iterations number of iterations performed by the Fisher-scoring algorithm.

estcoef a data frame with the estimated model coefficients in the first column (beta), their asymptotic standard errors in the second column (std.error), the t statistics in the third column (tvalue) and the p-values of the significance of each coefficient in last column (pvalue).

refvar estimated random effects variance.

goodness vector containing several goodness-of-fit measures: loglikehood, AIC, BIC, KIC and the measures proposed by Marhuenda et al. (2014): AICc, AICb1, AICb2, KICc, KICb1, KICb2. B must be must be greater than 0 to obtain these last measures.

In case that formula or vardir contain NA values a message is printed and no action is done.

References

- Fay, R.E. and Herriot, R.A. (1979). Estimation of income from small places: An application of James-Stein procedures to census data. Journal of the American Statistical Association 74, 269-277.

- Marhuenda, Y., Morales, D. and Pardo, M.C. (2014). Information criteria for Fay-Herriot model selection. Computational Statistics and Data Analysis 70, 268-280.

- Rao, J.N.K. (2003). Small Area Estimation. Wiley, London.

See Also

mseFH

Examples

```
# Load data set
data(milk)
attach(milk)

# Fit FH model using REML method with indicators of 4 Major Areas as
# explanatory variables.
resultREML <- eblupFH(yi ~ as.factor(MajorArea), SD^2)
resultREML

#Fit FH model using FH method
resultFH <- eblupFH(yi ~ as.factor(MajorArea), SD^2, method="FH")
resultFH

detach(milk)
```

---

eblupSFH *EBLUPs based on a spatial Fay-Herriot model.*

Description

This function gives small area estimators based on a spatial Fay-Herriot model, where area effects follow a SAR(1) process. Fitting method can be chosen between REML and ML.

Usage

```
eblupSFH(formula, vardir, proxmat, method = "REML", MAXITER = 100,
         PRECISION = 0.0001, data)
```

Arguments

formula	an object of class formula (or one that can be coerced to that class): a symbolic description of the model to be fitted. The variables included in formula must have a length equal to the number of domains D. Details of model specification are given under Details.
vardir	vector containing the D sampling variances of direct estimators for each domain. The values must be sorted as the variables in formula.
proxmat	D*D proximity matrix or data frame with values in the interval [0,1] containing the proximities between the row and column domains. The rows add up to 1. The rows and columns of this matrix must be sorted as the elements in formula.
method	type of fitting method, to be chosen between "REML" or "ML". Default value is REML.
MAXITER	maximum number of iterations allowed for the Fisher-scoring algorithm. Default value is 100.
PRECISION	convergence tolerance limit for the Fisher-scoring algorithm. Default value is 0.0001.
data	optional data frame containing the variables named in formula and vardir. By default the variables are taken from the environment from which eblupSHF is called.

Details

A typical model has the form response ~ terms where response is the (numeric) response vector and terms is a series of terms which specifies a linear predictor for response. A terms specification of the form first + second indicates all the terms in first together with all the terms in second with duplicates removed.

A formula has an implied intercept term. To remove this use either y ~ x - 1 or y ~ 0 + x. See formula for more details of allowed formulae.

Value

The function returns a list with the following objects:

eblup	vector with the values of the estimators for the domains.
fit	a list containing the following objects:

method type of fitting method applied ("REML" or "ML").

convergence a logical value equal to TRUE if Fisher-scoring algorithm converges in less than MAXITER iterations.

iterations number of iterations performed by the Fisher-scoring algorithm.

estcoef a data frame with the estimated model coefficients in the first column (beta), their asymptotic standard errors in the second column (std.error), the t statistics in the third column (tvalue) and the p-values of the significance of each coefficient in last column (pvalue).

refvar estimated random effects variance.

spatialcorr estimated spatial correlation parameter.

goodness vector containing three goodness-of-fit measures: loglikehood, AIC and BIC.

In case that formula, vardir or proxmat contain NA values a message is printed and no action is done.

Author(s)

Isabel Molina, Monica Pratesi and Nicola Salvati.

References

- Small Area Methods for Poverty and Living Conditions Estimates (SAMPLE), funded by European Commission, Collaborative Project 217565, Call identifier FP7-SSH-2007-1.

- Molina, I., Salvati, N. and Pratesi, M. (2009). Bootstrap for estimating the MSE of the Spatial EBLUP. Computational Statistics 24, 441-458.

- Petrucci, A. and Salvati, N. (2006). Small area estimation for spatial correlation in watershed erosion assessment. Journal of Agricultural, Biological and Environmental Statistics 11, 169-182.

- Pratesi, M. and Salvati, N. (2008). Small area estimation: the EBLUP estimator based on spatially correlated random area effects. Statistical Methods & Applications 17, 113-141.

See Also

mseSFH, npbmseSFH, pbmseSFH

Examples

```
data(grapes)       # Load data set
data(grapesprox)   # Load proximity matrix

# Fit Spatial Fay-Herriot model using ML method
resultML <- eblupSFH(grapehect ~ area + workdays - 1, var, grapesprox,
                     method="ML", data=grapes)
resultML

# Fit Spatial Fay-Herriot model using REML method
resultREML <- eblupSFH(grapehect ~ area + workdays - 1, var, grapesprox,
                       data=grapes)
resultREML
```

---

eblupSTFH *EBLUPs based on a spatio-temporal Fay-Herriot model.*

Description

Fits a spatio-temporal Fay-Herriot model with area effects following a SAR(1) process and with either uncorrelated or AR(1) time effects.

Usage

```
eblupSTFH(formula, D, T, vardir, proxmat, model = "ST", MAXITER = 100,
          PRECISION = 0.0001, theta_iter = FALSE,
          sigma21_start = 0.5 * median(vardir), rho1_start = 0.5,
          sigma22_start = 0.5 * median(vardir), rho2_start = 0.5,
          data)
```

Arguments

formula	an object of class formula (or one that can be coerced to that class): a symbolic description of the model to be fitted. The variables included in formula must have a length equal to D*T and sorted in the ascending order by the time instant within each domain. Details of model specification are given under Details.
D	total number of domains.
T	total number of time instants (constant for all domains).
vardir	vector containing the D*T sampling variances of direct estimators for each domain and time instant. The values must be sorted as the variables in formula.
proxmat	D*D proximity matrix or data frame with values in the interval [0,1] containing the proximities between the row and column domains. The rows add up to 1. The rows and columns of this matrix must be sorted by domain as the variables in formula.
model	type of model to be chosen between "ST" (AR(1) time-effects within each domain) or "S" (uncorrelated time effects within each domain). Default model is "ST".
MAXITER	maximum number of iterations allowed for the Fisher-scoring algorithm. Default value is 100.
PRECISION	convergence tolerance limit for the Fisher-scoring algorithm. Default value is 0.0001.
theta_iter	If TRUE the estimated values of area effects variance, area effects spatial autocorrelation, area-time effects variance and time autocorrelation parameter of the area-time effects of each iteration of the fitting algorithm are returned in estvarcomp_iterations variable.
sigma21_start	Starting value of the area effects variance in the fitting algorithm. Default value is 0.5*median(vardir).
rho1_start	Starting value of the area effects spatial autocorrelation parameter in the fitting algorithm. Default value is 0.5.
sigma22_start	Starting value of the area-time effects variance in the fitting algorithm. Default value is 0.5*median(vardir).
rho2_start	Starting value of the time autocorrelation parameter of the area-time effects in the fitting algorithm. Default value is 0.5.
data	optional data frame containing the variables named in formula and vardir. By default the variables are taken from the environment from which eblupSTFH is called.

Details

A typical model has the form response ~ terms where response is the (numeric) response vector and terms is a series of terms which specifies a linear predictor for response. A terms specification of the form first + second indicates all the terms in first together with all the terms in second with duplicates removed.

A formula has an implied intercept term. To remove this use either y ~ x - 1 or y ~ 0 + x. See formula for more details of allowed formulae.

Value

The function returns a list with the following objects:

`eblup` a column vector with length `D*T` with the values of the estimators for the `D` domains and `T` time instants.

`fit` a list containing the following objects:

- `model` type of model `"S"` or `"ST"`.
- `convergence` a logical value equal to `TRUE` if Fisher-scoring algorithm converges in less than `MAXITER` iterations.
- `iterations` number of iterations performed by the Fisher-scoring algorithm.
- `estcoef` a data frame with the estimated model coefficients in the first column (`beta`), their asymptotic standard errors in the second column (`std.error`), the t statistics in the third column (`tvalue`) and the p-values of the significance of each coefficient in last column (`pvalue`).
- `estvarcomp` a data frame with the estimated values of the variances and correlation coefficients in the first column (`estimate`) and their asymptotic standard errors in the second column (`std.error`).
- `estvarcomp_iterations:` if `theta_iter=TRUE`, this component contains a data frame with the estimated values of the variances and correlation coefficients obtained for each iteration of the fitting algorithm.
- `goodness` vector containing three goodness-of-fit measures: loglikehood, AIC and BIC.

In case that `formula`, `vardir` or `proxmat` contain NA values a message is printed and no action is done.

Author(s)

Yolanda Marhuenda, Isabel Molina and Domingo Morales.

References

- Small Area Methods for Poverty and Living Conditions Estimates (SAMPLE), funded by European Commission, Collaborative Project 217565, Call identifier FP7-SSH-2007-1.

- Marhuenda, Y., Molina, I. and Morales, D. (2013). Small area estimation with spatio-temporal Fay-Herriot models. Computational Statistics and Data Analysis 58, 308-325.

See Also

pbmseSTFH

Examples

```
data(spacetime)      # Load data set
data(spacetimeprox)  # Load proximity matrix

D <- nrow(spacetimeprox)            # number of domains
T <- length(unique(spacetime$Time)) # number of time instant

# Fit model S with uncorrelated time effects for each domain
resultS <- eblupSTFH(Y ~ X1 + X2, D, T, Var, spacetimeprox, "S",
                     theta_iter=TRUE, data=spacetime)
rowsT <- seq(T, T*D, by=T)
data.frame(Domain=spacetime$Area[rowsT], EBLUP_S=resultS$eblup[rowsT])
resultS$fit

# Fit model ST with AR(1) time effects for each domain
resultST <- eblupSTFH(Y ~ X1 + X2, D, T, Var, spacetimeprox,
                      theta_iter=TRUE, data=spacetime)
data.frame(Domain=spacetime$Area[rowsT], EBLUP_ST=resultS$eblup[rowsT])
resultST$fit
```

---

grapes *Synthetic data on grape production for the region of Tuscany.*

Description

Synthetic data on grape production with spatial correlation for 274 municipalities in the region of Tuscany.

Usage

```
data(grapes)
```

Format

A data frame with 274 observations on the following 4 variables.

`grapehect`: direct estimators of the mean agrarian surface area used for production of grape (in hectares) for each Tuscany municipality.

`area`: agrarian surface area used for production (in hectares).

`workdays`: average number of working days in the reference year (2000).

`var`: sampling variance of the direct estimators for each Tuscany municipality.

grapesprox *Proximity matrix for the spatial Fay-Herriot model.*

Description

A data frame containing the proximity values for the 274 municipalities in the region of Tuscany included in data set grapes.

Usage

```
data(grapesprox)
```

Format

The values are numbers in the interval [0,1] containing the proximity of the row and column domains. The sum of the values of each row is equal to 1.

incomedata *Synthetic income data.*

Description

Synthetic data on income and other related variables for Spanish provinces.

Usage

```
data(incomedata)
```

Format

A data frame with 17199 observations on the following 21 variables.

provlab: province name.

prov: province code.

ac: region of the province.

gen: gender: 1:male, 2:female.

age: age group: 0:<=13, 1:14-15, 2:16-24, 3:25-49, 4:50-64, 5: >=65.

nat: nationality: 1:Spanish, 2:other.

educ: education level: 0:age<16, 1:primary education (compulsory educ.), 2:secondary education, 3:post-secondary education.

labor: labor force status: 0:age<16, 1:employed, 2:unemployed, 3:inactive.

age2: indicator of age group 16-24.

age3: indicator of age group 25-49.

`age4`: indicator of age group 50-64.

`age5`: indicator of age group >=65.

`educ1`: indicator of education level 1 (primary education).

`educ2`: indicator of education level 2 (secondary education.

`educ3`: indicator of education level 3 (post-secondary education).

`nat1`: indicator of Spanish nationality.

`labor1`: indicator of being employed.

`labor2`: indicator of being unemployed.

`labor3`: indicator of being inactive.

`income`: normalized income.

`weight`: sampling weight.

`milk` *Data on fresh milk expenditure.*

Description

Data on fresh milk expenditure, used by Arora and Lahiri (1997) and by You and Chapman (2006).

Usage

```
data(milk)
```

Format

A data frame with 43 observations on the following 6 variables.

`SmallArea`: areas of inferential interest.

`ni`: sample sizes of small areas.

`yi`: average expenditure on fresh milk for the year 1989 (direct estimates for the small areas).

`SD`: estimated standard deviations of `yi`.

`CV`: estimated coefficients of variation of `yi`.

`MajorArea`: major areas created by You and Chapman (2006). These areas have similar direct estimates and produce a large CV reduction when using a FH model.

References

- Arora, V. and Lahiri, P. (1997). On the superiority of the Bayesian method over the BLUP in small area estimation problems. Statistica Sinica 7, 1053-1063.

- You, Y. and Chapman, B. (2006). Small area estimation using area level models and estimated sampling variances. Survey Methodology 32, 97-103.

mseFH *Mean squared error estimator of the EBLUP under a Fay-Herriot model.*

Description

Calculates the mean squared error estimator of the EBLUP under a Fay-Herriot model. The EBLUP might have been obtained by either ML, REML or by FH fitting methods.

Usage

```
mseFH(formula, vardir, method = "REML", MAXITER = 100, PRECISION = 0.0001, B = 0,
      data)
```

Arguments

formula	an object of class formula (or one that can be coerced to that class): a symbolic description of the model to be fitted. The variables included in formula must have a length equal to the number of domains D. Details of model specification are given under Details.
vardir	vector containing the D sampling variances of direct estimators for each domain. The values must be sorted as the variables in formula.
method	method used to fit the Fay-Herriot model, which can be either "ML", "REML" or "FH" methods. Default is "REML" method.
MAXITER	maximum number of iterations allowed in the Fisher-scoring algorithm. Default is 100 iterations.
PRECISION	convergence tolerance limit for the Fisher-scoring algorithm. Default value is 0.0001.
B	number of bootstrap replicates to calculate the goodness-of-fit measures proposed by Marhuenda et al. (2014). Default value is 0 indicating that these measures are not calculated.
data	optional data frame containing the variables named in formula and vardir. By default the variables are taken from the environment from which mseFH is called.

Details

A typical model has the form response ~ terms where response is the (numeric) response vector and terms is a series of terms which specifies a linear predictor for response. A terms specification of the form first + second indicates all the terms in first together with all the terms in second with duplicates removed.

A formula has an implied intercept term. To remove this use either y ~ x - 1 or y ~ 0 + x. See formula for more details of allowed formulae

Value

The function returns a list with the following objects:

`est`	a list with the results of the estimation process: `eblup` and `fit`. For the description of these objects, see Value of `eblupFH` function.
`mse`	a vector with the estimated mean squared errors of the EBLUPs for the small domains.

In case that `formula` or `vardir` contain NA values a message is printed and no action is done.

References

- Datta, G.S. and Lahiri, P. (2000). A unified measure of uncertainty of estimated best linear unbiased predictors in small area estimation problems. Statistica Sinica 10, 613-627.

- Datta, G.S., Rao, J.N.K. and Smith D.D. (2005). On measuring the variability of small area estimators under a basic area level model. Biometrika 92, 183-196.

- Fay, R.E. and Herriot, R.A. (1979). Estimation of income from small places: An application of James-Stein procedures to census data. Journal of the American Statistical Association 74, 269-277.

- Jiang, J. (1996). REML estimation: asymptotic behavior and related topics. Annals of Statistics 24, 255-286.

- Marhuenda, Y., Morales, D. and Pardo, M.C. (2014). Information criteria for Fay-Herriot model selection. Computational Statistics and Data Analysis 70, 268-280.

- Prasad, N. and Rao, J. (1990). The estimation of the mean squared error of small-area estimators. Journal of the American Statistical Association 85, 163-171.

See Also

`eblupFH`

Examples

```
# Load data set
data(milk)
attach(milk)

# Fit Fay-Herriot model using ML method with indicators
# of 4 Major Areas as explanatory variables and compute
# estimated MSEs of EB estimators
resultML <- mseFH(yi ~ as.factor(MajorArea), SD^2, method="ML")
resultML

# Fit Fay-Herriot model using REML method and compute
# estimated MSEs of EB estimators
resultREML <- mseFH(yi ~ as.factor(MajorArea), SD^2)
resultREML

# Fit Fay-Herriot model using FH method and compute
# estimated MSEs of EB estimators
resultFH <- mseFH(yi ~ as.factor(MajorArea), SD^2, method="FH")
```

```
resultFH

detach(milk)
```

---

mseSFH *Mean squared error estimator of the spatial EBLUP under a spatial Fay-Herriot model.*

Description

Calculates analytical mean squared error estimates of the spatial EBLUPs obtained from the fit of a spatial Fay-Herriot model, in which area effects follow a Simultaneously Autorregressive (SAR) process.

Usage

```
mseSFH(formula, vardir, proxmat, method = "REML", MAXITER = 100,
       PRECISION = 0.0001, data)
```

Arguments

formula	an object of class formula (or one that can be coerced to that class): a symbolic description of the model to be fitted. The variables included in formula must have a length equal to the number of domains D. Details of model specification are given under Details.
vardir	vector containing the D sampling variances of direct estimators for each domain. The values must be sorted as the variables in formula.
proxmat	D*D proximity matrix or data frame with values in the interval [0,1] containing the proximities between the row and column domains. The rows add up to 1. The rows and columns of this matrix must be sorted as the variables in formula.
method	type of fitting method, to be chosen between "REML" or "ML". Default value is REML.
MAXITER	maximum number of iterations allowed for the Fisher-scoring algorithm. Default value is 100.
PRECISION	convergence tolerance limit for the Fisher-scoring algorithm. Default value is 0.0001.
data	optional data frame containing the variables named in formula and vardir. By default the variables are taken from the environment from which mseSFH is called.

Value

The function returns a list with the following objects:

est	a list with the results of the estimation process: eblup and fit. For the description of these objects, see Value of eblupSFH function.

mse a vector with the analytical mean squared error estimates of the spatial EBLUPs.

In case that formula, vardir or proxmat contain NA values a message is printed and no action is done.

Author(s)

Isabel Molina, Monica Pratesi and Nicola Salvati.

References

- Small Area Methods for Poverty and Living Conditions Estimates (SAMPLE), funded by European Commission, Collaborative Project 217565, Call identifier FP7-SSH-2007-1.

- Molina, I., Salvati, N. and Pratesi, M. (2009). Bootstrap for estimating the MSE of the Spatial EBLUP. Computational Statistics 24, 441-458.

- Singh, B., Shukla, G. and Kundu, D. (2005). Spatio-temporal models in small area estimation. Survey Methodology 31, 183-195.

See Also

eblupSFH, npbmseSFH, pbmseSFH

Examples

```
data(grapes)       # Load data set
data(grapesprox)   # Load proximity matrix

# Calculate analytical MSE estimates using REML method
result <- mseSFH(grapehect ~ area + workdays - 1, var, grapesprox, data=grapes)
result
```

---

npbmseSFH *Nonparametric bootstrap mean squared error estimator of the spatial EBLUPs under a spatial Fay-Herriot model.*

Description

Calculates nonparametric bootstrap mean squared error estimates of the spatial EBLUPs obtained by fitting a spatial Fay-Herriot model, in which area effects follow a Simultaneously Autorregressive (SAR) process.

Usage

```
npbmseSFH(formula, vardir, proxmat, B = 100, method = "REML", MAXITER = 100,
          PRECISION = 0.0001, data)
```

Arguments

formula	an object of class formula (or one that can be coerced to that class): a symbolic description of the model to be fitted. The variables included in formula must have a length equal to the number of domains D. Details of model specification are given under Details.
vardir	vector containing the D sampling variances of direct estimators for each domain. The values must be sorted as the variables in formula.
proxmat	D*D proximity matrix or data frame with values in the interval [0,1] containing the proximities between the row and column domains. The rows add up to 1. The rows and columns of this matrix must be sorted as the variables in formula.
B	number of bootstrap replicates. Default value is 100.
method	type of fitting method. Currently only "REML" method is available.
MAXITER	maximum number of iterations allowed for the Fisher-scoring algorithm. Default value is 100.
PRECISION	convergence tolerance limit for the Fisher-scoring algorithm. Default value is 0.0001.
data	optional data frame containing the variables named in formula and vardir. By default the variables are taken from the environment from which npbmseSFH is called.

Details

This function uses random number generation. To fix the seed, use set.seed.

A typical model has the form response ~ terms where response is the (numeric) response vector and terms is a series of terms which specifies a linear predictor for response. A terms specification of the form first + second indicates all the terms in first together with all the terms in second with duplicates removed. A terms specification of the form first + second indicates all the terms in first together with all the terms in second with any duplicates removed.

A formula has an implied intercept term. To remove this use either y ~ x - 1 or y ~ 0 + x. See formula for more details of allowed formulae.

Value

The function returns a list with the following objects:

est	a list with the results of the estimation process: eblup and fit. For the description of these objects, see Value of eblupSFH function.
mse	data frame containing the naive nonparametric bootstrap mean squared error estimates of the spatial EBLUPs (mse) and the bias-corrected nonparametric bootstrap mean squared error estimates of the spatial EBLUPs (msebc).

In case that formula, vardir or proxmat contain NA values a message is printed and no action is done.

Author(s)

Isabel Molina, Monica Pratesi and Nicola Salvati.

References

- Small Area Methods for Poverty and Living Conditions Estimates (SAMPLE), funded by European Commission, Collaborative Project 217565, Call identifier FP7-SSH-2007-1.

- Molina, I., Salvati, N. and Pratesi, M. (2009). Bootstrap for estimating the MSE of the Spatial EBLUP. Computational Statistics 24, 441-458.

See Also

eblupSFH, pbmseSFH, mseSFH

Examples

```
data(grapes)       # Load data set
data(grapesprox)   # Load proximity matrix

# Obtain the naive and bias-corrected non parametric bootstrap MSE
# estimates using REML
set.seed(123)
result <- npbmseSFH(grapehect ~ area + workdays - 1, var, grapesprox, B=2, data=grapes)
result
```

---

pbmseBHF *Parametric bootstrap mean squared error estimators of the EBLUPs of means obtained under a nested error linear regression model.*

Description

Calculates, for selected domains, parametric bootstrap mean squared error estimators of the EBLUPs of means, when EBLUPs are obtained from a nested error linear regression model.

Usage

```
pbmseBHF(formula, dom, selectdom, meanxpop, popnsize, B = 200, method = "REML",
         data)
```

Arguments

formula	an object of class formula (or one that can be coerced to that class): a symbolic description of the model to be fitted. The details of model specification are given under Details.
dom	n*1 vector or factor (same size as y in formula) with domain codes.
selectdom	I*1 optional vector or factor with the domain codes for which we want to estimate the means. It must be a subset of the domain codes in dom. If this parameter is not included all the domain codes included in dom are considered.

meanxpop	D*(p+1) data frame with domain codes in the first column. Each remaining column contains the population means of each of the p auxiliary variables for the D domains. The domains considered in meanxpop must contain those specified in selectdom (D>=I).
popnsize	D*2 data frame with domain codes in the first column and the corresponding domain population sizes in the second column. The domains considered in popnsize must contain those specified in selectdom (D>=I).
B	number of bootstrap replicates. Default is 50.
method	a character string. If "REML" the model is fitted by maximizing the restricted log-likelihood. If "ML" the log-likelihood is maximized. Defaults to "REML".
data	optional data frame containing the variables named in formula and dom. By default the variables are taken from the environment from which pbmseBHF is called.

Details

This function uses random number generation. To fix the seed, use set.seed.

A typical model has the form response ~ terms where response is the (numeric) response vector and terms is a series of terms which specifies a linear predictor for response. A terms specification of the form first + second indicates all the terms in first together with all the terms in second with duplicates removed.

A formula has an implied intercept term. To remove this use either y ~ x - 1 or y ~ 0 + x. See formula for more details of allowed formulae.

Value

The function returns a list with the following objects:

est	a list with the results of the estimation process: eblup and fit. For the description of these objects, see Value of eblupBHF function.
mse	data frame with number of rows equal to number of selected domains, containing in its columns the domain codes (domain) and the parametric bootstrap mean squared error estimators (mse).

Cases with NA values in formula or dom are ignored.

References

- Gonzalez-Manteiga, W., Lombardia, M., Molina, I., Morales, D. and Santamaria, L. (2008). Analytic and bootstrap approximations of prediction errors under a multivariate Fay-Herriot model. Computational Statistics and Data Analysis 52, 5242-5252.

- Molina, I. and Rao, J.N.K. (2010). Small Area Estimation of Poverty Indicators. The Canadian Journal of Statistics 38, 369-385.

See Also

eblupBHF

Examples

```
# Load data set for segments (units within domains)
data(cornsoybean)

# Load data set for counties
data(cornsoybeanmeans)
attach(cornsoybeanmeans)

# Construct data frame with county means of auxiliary variables for
# domains. First column must include the county code
Xmean <- data.frame(CountyIndex, MeanCornPixPerSeg, MeanSoyBeansPixPerSeg)
Popn  <- data.frame(CountyIndex, PopnSegments)

# Compute parametric bootstrap MSEs of the EBLUPs of means of crop areas
# for each county.
set.seed(123)
result <- pbmseBHF(CornHec ~ CornPix + SoyBeansPix, dom=County,
                   selectdom=c(10,1,5), meanxpop=Xmean, popnsize=Popn,
                   B=50, data=cornsoybean)
result

detach(cornsoybeanmeans)
```

---

pbmseebBHF *Parametric bootstrap mean squared error estimators of EB estimators.*

Description

This function obtains estimators of the mean squared errors of the EB estimators of domain parameters by a parametric bootstrap method. Population values of auxiliary variables are required.

Usage

```
pbmseebBHF(formula, dom, selectdom, Xnonsample, B = 100, MC = 100, data,
           transform = "BoxCox", lambda = 0, constant = 0, indicator)
```

Arguments

formula	an object of class formula (or one that can be coerced to that class): a symbolic description of the model to be fitted. The details of model specification are given under Details.
dom	n*1 vector or factor (same size as y in formula) with domain codes.
selectdom	I*1 optional vector or factor with the domain codes for which we want to estimate the indicators. It must be a subset of the domain codes in dom. If this parameter is not included, the unique domain codes included in dom are considered.

Xnonsample	matrix or data frame containing in the first column the domain codes and in the rest of columns the values of each of p auxiliary variables for the out-of-sample units in each selected domain.
B	number of bootstrap replicates. Default value is 100.
MC	number of Monte Carlo replicates for the empirical approximation of the EB estimator. Default value is 100.
data	optional data frame containing the variables named in formula and dom. By default the variables are taken from the environment from which pbmseebBHF is called.
transform	type of transformation for the dependent variable to be chosen between the "BoxCox" and "power" families so that the dependent variable in formula follows approximately a Normal distribution. Default value is "BoxCox".
lambda	value for the parameter of the family of transformations specified in transform. Default value is 0, which gives the log transformation for the two possible families.
constant	constant added to the dependent variable before doing the transformation, to achieve a distribution close to Normal. Default value is 0.
indicator	function of the (untransformed) variable on the left hand side of formula that we want to estimate in each domain.

Details

This function uses random number generation. To fix the seed, use set.seed.

A typical model has the form response ~ terms where response is the (numeric) response vector and terms is a series of terms which specifies a linear predictor for response. A terms specification of the form first + second indicates all the terms in first together with all the terms in second with duplicates removed. A terms specification of the form first + second indicates all the terms in first together with all the terms in second with any duplicates removed.

A formula has an implied intercept term. To remove this use either y ~ x - 1 or y ~ 0 + x. See formula for more details of allowed formulae.

Value

The function returns a list with the following objects:

est	a list with the results of the estimation process: eb and fit. For the description of these objects, see Value of ebBHF function.
mse	data frame with number of rows equal to number of selected domains, containing in its columns the domain codes (domain) and the parametric bootstrap mean squared error estimates of indicator (mse).

Cases with NA values in formula or dom are ignored.

References

- Small Area Methods for Poverty and Living Conditions Estimates (SAMPLE), funded by European Commission, Collaborative Project 217565, Call identifier FP7-SSH-2007-1.

- Molina, I. and Rao, J.N.K. (2010). Small Area Estimation of Poverty Indicators. The Canadian Journal of Statistics 38, 369-385.

See Also

ebBHF

Examples

```
data(incomedata)         # Load data set
attach(incomedata)

# Construct design matrix for sample elements
Xs<-cbind(age2,age3,age4,age5,nat1,educ1,educ3,labor1,labor2)

# Select the domains to compute EB estimators
data(Xoutsamp)
domains <- c(5)

# Poverty incidence indicator
povertyline <- 0.6*median(incomedata$income)
povertyline                          # 6477.484
povinc <- function(y)
{
   z <- 6477.484
   result <- mean(y<z)
   return (result)
}

# Compute parametric bootstrap MSE estimators of the EB
# predictors of poverty incidence. Take constant=3600 to achieve
# approximately symmetric residuals.
set.seed(123)
result <- pbmseebBHF(income~Xs, dom=prov, selectdom=domains,
                     Xnonsample=Xoutsamp, B=2, MC=2, constant=3600,
                     indicator=povinc)
result$est$eb
result$mse
result$est$fit$refvar

detach(incomedata)
```

---

pbmseSFH *Parametric bootstrap mean squared error estimators of the spatial EBLUPs under a spatial Fay-Herriot model.*

Description

Calculates the parametric bootstrap mean squared error estimates of the spatial EBLUPs obtained by fitting the spatial Fay-Herriot model, in which area effects follow a Simultaneously Autorregressive (SAR) process.

Usage

```
pbmseSFH(formula, vardir, proxmat, B = 100, method = "REML", MAXITER = 100,
         PRECISION = 0.0001, data)
```

Arguments

formula	an object of class formula (or one that can be coerced to that class): a symbolic description of the model to be fitted. The variables included in formula must have a length equal to the number of domains D. Details of model specification are given under Details.
vardir	vector containing the D sampling variances of direct estimators for each domain. The values must be sorted as the variables in formula.
proxmat	D*D proximity matrix or data frame with values in the interval [0,1] containing the proximities between the row and column domains. The rows add up to 1. The rows and columns of this matrix must be sorted as the variables in formula.
B	number of bootstrap replicates. Default value is 100.
method	type of fitting method, to be chosen between "REML" or "ML". Default value is REML.
MAXITER	maximum number of iterations allowed for the Fisher-scoring algorithm. Default value is 100.
PRECISION	convergence tolerance limit for the Fisher-scoring algorithm. Default value is 0.0001.
data	optional data frame containing the variables named in formula and vardir. By default the variables are taken from the environment from which pbmseSFH is called.

Details

This function uses random number generation. To fix the seed, use set.seed.

A typical model has the form response ~ terms where response is the (numeric) response vector and terms is a series of terms which specifies a linear predictor for response. A terms specification of the form first + second indicates all the terms in first together with all the terms in second with duplicates removed. A terms specification of the form first + second indicates all the terms in first together with all the terms in second with any duplicates removed.

A formula has an implied intercept term. To remove this use either y ~ x - 1 or y ~ 0 + x. See formula for more details of allowed formulae.

Value

The function returns a list with the following objects:

`est`	a list with the results of the estimation process: `eblup` and `fit`. For the description of these objects, see Value of `eblupSFH` function.
`mse`	data frame containing the naive parametric bootstrap mean squared error estimates (`mse`) and the bias-corrected parametric bootstrap mean squared error estimates of the spatial EBLUPs (`msebc`).

In case that `formula`, `vardir` or `proxmat` contain NA values a message is printed and no action is done.

Author(s)

Isabel Molina, Monica Pratesi and Nicola Salvati.

References

- Small Area Methods for Poverty and Living Conditions Estimates (SAMPLE), funded by European Commission, Collaborative Project 217565, Call identifier FP7-SSH-2007-1.

- Molina, I., Salvati, N. and Pratesi, M. (2009). Bootstrap for estimating the MSE of the Spatial EBLUP. Computational Statistics 24, 441-458.

See Also

`eblupSFH`, `npbmseSFH`, `mseSFH`

Examples

```
data(grapes)       # Load data set
data(grapesprox)   # Load proximity matrix

# Obtain the fitting values, naive and bias-corrected parametric bootstrap MSE estimates
# using REML method
set.seed(123)
result <- pbmseSFH(grapehect ~ area + workdays - 1, var, grapesprox, B=2, data=grapes)
result
```

---

pbmseSTFH *Parametric bootstrap mean squared error estimator of a spatio-temporal Fay-Herriot model.*

Description

Calculates parametric bootstrap mean squared error estimates of the EBLUPs based on a spatio-temporal Fay-Herriot model with area effects following a SAR(1) process and with either uncorrelated or correlated time effects within each domain following an AR(1) process.

Usage

```
pbmseSTFH(formula, D, T, vardir, proxmat, B = 100, model = "ST",
          MAXITER = 100, PRECISION = 0.0001, theta_iter = FALSE,
          sigma21_start = 0.5 * median(vardir), rho1_start = 0.5,
          sigma22_start = 0.5 * median(vardir), rho2_start = 0.5,
          data)
```

Arguments

formula	an object of class formula (or one that can be coerced to that class): a symbolic description of the model to be fitted. The variables included in formula must have a length equal to D*T and sorted in the ascending order by the time instant within each domain. Details of model specification are given under Details.
D	total number of domains.
T	total number of time instants (constant for each domain).
vardir	vector containing the n=D*T sampling variances for each domain and time instant. The values must be sorted as the variables in formula.
proxmat	D*D proximity matrix or data frame with values in the interval [0,1] containing the proximities between the row and column domains. The rows add up to 1. The rows and columns of this matrix must be sorted by domain as the variables in formula.
B	number of bootstrap replicates. Default value is 100.
model	type of model to be chosen between "ST" (correlated time-effects within domains) or "S" (uncorrelated time-effects within domains).
MAXITER	maximum number of iterations allowed for the Fisher-scoring algorithm. Default value is 100.
PRECISION	convergence tolerance limit for the Fisher-scoring algorithm. Default value is 0.0001.
theta_iter	If TRUE the estimated values of area effects variance, area effects spatial autocorrelation, area-time effects variance and time autocorrelation parameter of the area-time effects of each iteration of the fitting algorithm are returned in est$estvarcomp_iterations.
sigma21_start	Starting value of the area effects variance in the fitting algorithm. Default value is 0.5*median(vardir).
rho1_start	Starting value of the area effects spatial autocorrelation parameter in the fitting algorithm. Default value is 0.5.
sigma22_start	Starting value of the area-time effects variance in the fitting algorithm. Default value is 0.5*median(vardir).
rho2_start	Starting value of the time autocorrelation parameter of the area-time effects in the fitting algorithm. Default value is 0.5.
data	optional data frame containing the variables named in formula and vardir. By default the variables are taken from the environment from which pbmseSTFH is called.

Details

This function uses random number generation. To fix the seed, use `set.seed`.

A typical model has the form response ~ terms where response is the (numeric) response vector and terms is a series of terms which specifies a linear predictor for response. A terms specification of the form first + second indicates all the terms in first together with all the terms in second with duplicates removed.

A formula has an implied intercept term. To remove this use either y ~ x - 1 or y ~ 0 + x. See `formula` for more details of allowed formulae.

Value

The function returns a list with the following objects:

`est`	a list with the results of the estimation process: `eblup` and `fit`. For the description of these objects, see Value of `eblupSTFH` function.
`mse`	a vector of length `D*T` containing the parametric bootstrap mean squared error estimates for the `D` domains and `T` time instants.

In case that `formula`, `vardir` or `proxmat` contain NA values a message is printed and no action is done.

Author(s)

Yolanda Marhuenda, Isabel Molina and Domingo Morales.

References

- Small Area Methods for Poverty and Living Conditions Estimates (SAMPLE), funded by European Commission, Collaborative Project 217565, Call identifier FP7-SSH-2007-1.

- Marhuenda, Y., Molina, I. and Morales, D. (2013). Small area estimation with spatio-temporal Fay-Herriot models. Computational Statistics and Data Analysis 58, 308-325.

See Also

`eblupSTFH`

Examples

```
data(spacetime)      # Load data set
data(spacetimeprox)  # Load proximity matrix

D <- nrow(spacetimeprox)            # number of domains
T <- length(unique(spacetime$Time)) # number of time instant

# Calculate MSEs of EBLUPs under the spatio-temporal Fay-Herriot model
# with uncorrelated time effects nested within domains (model S)
set.seed(123)
resultS <- pbmseSTFH(Y ~ X1 + X2, D, T, Var, spacetimeprox, B=10,
                     model="S", data=spacetime)
```

```
# Print direct estimates, variance, "S" model estimates, mse and
# residuals of the last time instant.
output <- data.frame(Domain=spacetime$Area, Period=spacetime$Time,
                     Direct=spacetime$Y, EBLUP_S=resultS$est$eblup,
                     VarDirect=spacetime$Var, MSE_S=resultS$mse,
                     Residuals=spacetime$Y-resultS$est$eblup)
periods <- unique(spacetime$Time)
lastperiod <- periods[length(periods)]
print(output[output[,"Period"]==lastperiod,], row.names=FALSE)


# Calculate MSEs of the EBLUPs based on the spatio-temporal Fay-Herriot model
# with AR(1) time effects nested within each area
attach(spacetime)
set.seed(123)
resultST <- pbmseSTFH(Y ~ X1 + X2, D, T, vardir=Var, spacetimeprox, B=10)

# Print direct estimates, variance, "ST" model estimates, mse and
# residuals of the last time instant.
output <- data.frame(Domain=Area, Period=Time, Direct=Y,
                     EBLUP_ST=resultST$est$eblup, VarDirect=Var,
                     MSE_ST=resultST$mse,
                     Residuals=Y-resultST$est$eblup)
periods <- unique(Time)
lastperiod <- periods[length(periods)]
print(output[output[,"Period"]==lastperiod,], row.names=FALSE)

detach(spacetime)
```

`pssynt` *Post-stratified synthetic estimators of domain means.*

Description

Calculates post-stratified synthetic estimators of domain means using the categories of a cualitative variable as post-strata.

Usage

```
pssynt(y, sweight, ps, domsizebyps, data)
```

Arguments

`y`	vector specifying the individual values of the variable for which we want to estimate the domain means.
`sweight`	vector (same size as `y`) with the sampling weights of the units.
`ps`	vector (same size as `y`) of factor with post-strata codes.

domsizebyps	data frame with domain codes in the first column. Each remaining column contains the domain population sizes for each post-strata. Names of these columns must be the post-strata identifiers specified in ps.
data	optional data frame containing the variables named in y, sweight and ps. By default the variables are taken from the environment from which pssynt is called.

Value

The function returns a data frame of size D*2 with the following columns:

Domain	domain codes in ascending order.
PsSynthetic	post-stratified synthetic estimators of domain means of variable y.

Cases with NA values in y, sweight or ps are ignored.

References

- Rao, J.N.K. (2003). "Small Area Estimation". Wiley, London.

See Also

direct, ssd

Examples

```
# Compute post-stratified synthetic estimators of mean income
# for provinces considering the education levels codes
# (variable educ) as post-strata.

# Load data set
data(incomedata)

# Load province sizes by education levels
data(sizeprovedu)

# Compute post-stratified synthetic estimators with province labels
# as domain codes
colnames(sizeprovedu) <- c("provlab", "prov", "0", "1", "2", "3")
result1 <- pssynt(y=income, sweight=weight, ps=educ,
                  domsizebyps=sizeprovedu[,-2], data=incomedata)
result1

# Now with province codes as domain codes
colnames(sizeprovedu) <- c("provlab", "prov", "0", "1", "2", "3")
result2 <- pssynt(y=income, sweight=weight, ps=educ,
                  domsizebyps=sizeprovedu[,-1], data=incomedata)
result2
```

sizeprov *Domain population sizes.*

Description

Identifiers and population sizes for domains in data set incomedata.

Usage

```
data(sizeprov)
```

Format

A data frame with 52 observations on the following 3 variables.

provlab: province name.

prov: province code.

Nd: province population count.

sizeprovage *Domain population sizes by age.*

Description

Names, codes and population sizes by age for domains in data set incomedata.

Usage

```
data(sizeprovage)
```

Format

A data frame with 52 observations on the following 7 variables.

provlab: province name.

prov: province code.

age1: province count for age group <16.

age2: province count for age group 16-24.

age3: province count for age group 25-49.

age4: province count for age group 50-64.

age5: province count for age group >=65.

sizeprovedu *Domain population sizes by level of education.*

Description

Identifiers and population sizes by level of education for domains in data set incomedata.

Usage

```
data(sizeprovedu)
```

Format

A data frame with 52 observations on the following 6 variables.

provlab: province name.

prov: province code.

educ0: province count for education level 0 (age<16).

educ1: province count for education level 1 (primary education).

educ2: province count for education level 2 (secondary education).

educ3: province count for education level 3 (post-secondary education).

sizeprovlab *Domain population sizes by labor force status.*

Description

Names, codes and population sizes by labor force status for domains in data set incomedata.

Usage

```
data(sizeprovlab)
```

Format

A data frame with 52 observations on the following 6 variables.

provlab: province name.

prov: province code.

labor0 province count for labor force status 0 (age<16).

labor1 province count for labor force status 1 (employed).

labor2 province count for labor force status 2 (unemployed).

labor3 province count for labor force status 3 (inactive).

sizeprovnat *Domain population sizes for Spanish or non Spanish nationality.*

Description

Names, codes and population sizes for Spanish or non Spanish nationality for domains in data set incomedata.

Usage

```
data(sizeprovnat)
```

Format

A data frame with 52 observations on the following 4 variables.

`provlab`: province name.

`prov`: province code.

`nat1`: province count for Spanish nationality.

`nat2`: province count for non Spanish nationality.

spacetime *Synthetic area level data with spatial and temporal correlation.*

Description

Synthetic area level data with spatial and temporal correlation.

Usage

```
data(spacetime)
```

Format

A data frame with 33 observations on the following 6 variables.

`Area`: numeric domain indicator.

`Time`: numeric time instant indicator.

`X1`: first auxiliary variable at domain level.

`X2`: second auxiliary variable at domain level.

`Y`: direct estimators of the target variable in the domains.

`Var`: sampling variances of direct estimators for each domain.

spacetimeprox *Proximity matrix for the spatio-temporal Fay-Herriot model.*

Description

Example of proximity matrix for the domains included in data set spacetime.

Usage

```
data(spacetimeprox)
```

Format

The values are numbers in the interval [0,1] containing the proximity of the row and column domains. The sum of the values of each row is equal to 1.

ssd *Sample size dependent estimator.*

Description

Calculates sample size dependent estimators of domain means, as composition of direct and synthetic estimators. The estimators involved in the composition must be given as function arguments.

Usage

```
ssd(dom, sweight, domsize, direct, synthetic, delta = 1, data)
```

Arguments

dom	vector or factor (same size as y) with domain codes.
sweight	vector (same size as dom) with sampling weights of the units.
domsize	matrix or data frame with domain codes in the first column and the corresponding domain population sizes in the second column.
direct	matrix or data frame with domain codes in the first column and the corresponding direct estimators of domain means in the second column.
synthetic	matrix or data frame with domain codes in the first column and the corresponding synthetic estimators of domain means in the second column.
delta	constant involved in sample size dependent estimator, controlling how much strength to borrow. Default value is 1.
data	optional data frame containing the variables named in dom and sweight. By default the variables are taken from the environment from which ssd is called.

Value

The function returns a data frame of size `D*2` with the following columns:

`Domain`	domain codes in ascending order.
`ssd`	sample size dependent estimators of domain means.
`CompWeight`	weights attached to direct estimators in the composition.

Cases with NA values in `dom` or `sweight` are ignored.

References

- Drew, D., Singh, M.P. and Choudhry, G.H. (1982). Evaluation of small area estimation techniques for the Canadian Labour Force Survey. Survey Methodology 8, 17-47.

- Rao, J. N. K. (2003). Small Area Estimation. Wiley, London.

See Also

`direct`, `pssynt`

Examples

```
# We compute sample size dependent estimators of mean income by
# composition of the Horvitz-Thompson direct estimator and the
# post-stratified synthetic estimator with age groups as post-strata.

# Load data set
data(incomedata)

# Load population sizes of provinces (domains)
data(sizeprov)

# First we compute Horvitz-Thompson direct estimators
dir <- direct(y=income, dom=provlab, sweight=weight,
              domsize=sizeprov[,c(1,3)], data=incomedata)

# Now we compute post-stratified synthetic estimators with education
# levels as post-strata
# Load province sizes by education levels
data(sizeprovedu)

# Compute post-stratified synthetic estimators
colnames(sizeprovedu) <- c("provlab", "prov", "0", "1", "2", "3")
synth <- pssynt(y=income, sweight=weight, ps=educ,
                domsizebyps=sizeprovedu[,-2], data=incomedata)

# Compute sample size dependent estimators of province mean income
# by composition of Horvitz-Thompson direct estimators and
# post-stratified estimators for delta=1
comp <- ssd(dom=provlab, sweight=weight, domsize=sizeprov[,c(1,3)],
            direct=dir[,c("Domain","Direct")], synthetic=synth, data=incomedata)
comp
```

Xoutsamp *Out-of-sample values of auxiliary variables for 5 domains.*

Description

Values of p auxiliary variables for out-of-sample units within 5 domains of data set incomedata.

Usage

```
data(Xoutsamp)
```

Format

A data frame with 713301 observations on the following 10 variables.

domain: a numeric vector with the domain codes.

age2: indicator of age group 16-24.

age3: indicator of age group 25-49.

age4: indicator of age group 50-64.

age5: indicator of age group >=65.

nat1: indicator of Spanish nationality.

educ1: indicator of education level 1 (primary education).

educ3: indicator of education level 3 (post-secondary education).

labor1: indicator of being employed.

labor2: indicator of being unemployed.

Index

∗ **datasets**
cornsoybean, 5
cornsoybeanmeans, 6
grapes, 20
grapesprox, 21
incomedata, 21
milk, 22
sizeprov, 39
sizeprovage, 39
sizeprovedu, 40
sizeprovlab, 40
sizeprovnat, 41
spacetime, 41
spacetimeprox, 42
Xoutsamp, 44
∗ **method**
bxcx, 4
diagonalizematrix, 6
direct, 7
ebBHF, 9
eblupBHF, 11
eblupFH, 13
eblupSFH, 15
eblupSTFH, 17
mseFH, 23
mseSFH, 25
npbmseSFH, 26
pbmseBHF, 28
pbmseebBHF, 30
pbmseSFH, 32
pbmseSTFH, 34
pssynt, 37
ssd, 42

bxcx, 4

cornsoybean, 5, *6*
cornsoybeanmeans, 6

diagonalizematrix, 6

direct, 7, *38*, *43*

ebBHF, 9, *31, 32*
eblupBHF, 11, *29*
eblupFH, 13, *24*
eblupSFH, 15, *25–28*, *34*
eblupSTFH, 17, *36*

formula, *9–12*, *14*, *16*, *18, 19*, *23*, *25*, *27–31*, *33*, *35, 36*

grapes, 20, *21*
grapesprox, 21

incomedata, 21, *39–41*, *44*

milk, 22
mseFH, *15*, 23
mseSFH, *17*, 25, *28*, *34*

npbmseSFH, *17*, *26*, 26, *34*

pbmseBHF, *13*, 28
pbmseebBHF, *10*, 30
pbmseSFH, *17*, *26*, *28*, 32
pbmseSTFH, *20*, 34
pssynt, *8*, 37, *43*

sae *(*sae-package*)*, 2
sae-package, 2
sizeprov, 39
sizeprovage, 39
sizeprovedu, 40
sizeprovlab, 40
sizeprovnat, 41
spacetime, 41, *42*
spacetimeprox, 42
ssd, *8*, *38*, 42

Xoutsamp, 44