Package ‘msaeDB’

April 8, 2021

Type Package
Title Difference Benchmarking for Multivariate Small Area Estimation
Version 0.2.1
Author Zaza Yuda Perwira, Azka Ubaidillah
Maintainer Zaza Yuda Perwira <221710086@stis.ac.id>
Description Implements Benchmarking Method for Multivariate Small Area Estimation under Fay Herriot Model. Multivariate Small Area Estimation (MSAE) is a development of Univariate Small Area Estimation that considering the correlation among response variables and borrowing the strength from related areas and auxiliary variables to increase the effectiveness of sample size, the multivariate model in this package is based on multivariate model 1 proposed by Roberto Be-navent and Domingo Morales (2016) <doi:10.1016/j.csda.2015.07.013>. Benchmarking in Small Area Estimation is a modification of Small Area Estimation model to guarantee that the aggregate weighted mean of the county predictors equals the corresponding weighted mean of survey estimates. Difference Benchmarking is the simplest benchmarking method but widely used by multiplying empirical best linear unbiased prediction (EBLUP) estimator by the common adjustment factors (J.N.K Rao and Isabel Molina, 2015).
License GPL-3
Encoding UTF-8
LazyData true
RoxygenNote 7.1.1
URL https://github.com/zazaperwira/msaeDB
BugReports https://github.com/zazaperwira/msaeDB/issues
Suggests knitr, rmarkdown, covr
VignetteBuilder knitr
Imports MASS, magic, stats
Depends R (>= 2.10)
NeedsCompilation no
Repository CRAN
Date/Publication 2021-04-08 10:50:03 UTC
R topics documented:

datamsaeDB ................................................................. 2
datamsaeDBns ............................................................ 3
msaedb ................................................................. 4
msaebdns ............................................................. 6
msaefh ................................................................. 9
msaefhns ............................................................ 10
saedb ............................................................... 12
saedbnns .......................................................... 14
saefh ............................................................ 17
saefhns ............................................................ 18

Index 21

datamsaeDB  Sample Data for Multivariate Small Area Estimation with Difference Benchmarking

Description

Dataset to simulate difference benchmarking of Multivariate Fay Herriot model

This data is generated base on multivariate Fay Herriot model by these following steps:

1. Generate explanatory variables $X_1$ and $X_2$. Take $\mu_{X_1} = \mu_{X_2} = 10$, $\sigma_{X_1} = 1$, $\sigma_{X_2} = 2$, and $\rho_x = 1/2$.
   Sampling error $e$ is generated with the following $\sigma_{e11} = 0.15$, $\sigma_{e22} = 0.25$, $\sigma_{e33} = 0.35$, and $\rho_e = 1/2$.
   For random effect $u$, we set $\sigma_{u11} = 0.2$, $\sigma_{u22} = 0.6$, and $\sigma_{u33} = 1.8$.
   For the weight we generate $w_1 \sim U(25,30)$, $w_2 \sim U(25,30)$, $w_3 \sim U(25,30)$
   Calculate direct estimation $Y_1 \ Y_2 \ Y_3$ where $Y_i = X_i \beta + u_i + e_i$

2. Then combine the direct estimations $Y_1 \ Y_2 \ Y_3$, explanatory variables $X_1 \ X_2$, weights $w_1 \ w_2 \ w_3$, and sampling varians covarians $v1 \ v12 \ v13 \ v2 \ v23 \ v3$ in a dataframe then named as datamsaeDB

Usage

datamsaeDB

Format

A data frame with 30 rows and 14 variables:

Y1 Direct Estimation of Y1
Y2 Direct Estimation of Y2
Y3 Direct Estimation of Y3
X1 Auxiliary variable of X1
**datamsaeDBns**

**X2**  Auxiliary variable of X2  
**w1**  Known proportion of units in small areas of Y1  
**w2**  Known proportion of units in small areas of Y2  
**w3**  Known proportion of units in small areas of Y3  
**v1**  Sampling Variance of Y1  
**v12**  Sampling Covariance of Y1 and Y2  
**v13**  Sampling Covariance of Y1 and Y3  
**v2**  Sampling Variance of Y2  
**v23**  Sampling Covariance of Y2 and Y3  
**v3**  Sampling Variance of Y3

---

**Description**

Dataset to simulate difference benchmarking of Multivariate Fay Herriot model for non-sampled area using clustering This data is generated base on multivariate Fay Herriot model by these following steps:

1. Generate explanatory variables $X_1$ and $X_2$. Take $\mu_{X_1} = \mu_{X_2} = 10$, $\sigma_{X_11}=1$, $\sigma_{X_2}=2$, and $\rho_x = 1/2$.  
   Sampling error $e$ is generated with the following $\sigma_{e11} = 0.15$, $\sigma_{e22} = 0.25$, $\sigma_{e33} = 0.35$, and $\rho_e = 1/2$.  
   For random effect $u$, we set $\sigma_{u11}=0.2$, $\sigma_{u22}=0.6$, and $\sigma_{u33}=1.8$.  
   For the weight we generate $w1$ $w2$ $w3$ by set the $w1 \sim U(25, 30)$, $w2 \sim U(25, 30)$, $w3 \sim U(25, 30)$  
   Calculate direct estimation $Y_1$ $Y_2$ $Y_3$ where $Y_i = X_i \beta + u_i + e_i$  
   c11 c12 c13 were obtained using K-Means clustering from the explanatory variables.

2. Then combine the direct estimations $Y_1$ $Y_2$ $Y_3$, explanatory variables $X_1$ $X_2$, weights $w1$ $w2$ $w3$, and sampling varians covarians $v1$ $v12$ $v13$ $v2$ $v23$ $v3$ in a data frame then named as datamsaeDB

**Usage**

datamsaeDBns

**Format**

A data frame with 30 rows and 18 variables:

- **clY1**  cluster information of Y1  
- **clY2**  cluster information of Y2  
- **clY3**  cluster information of Y3
**nonsample**  A column with logical values, TRUE if the area is non-sampled

**Y1**  Direct Estimation of Y1

**Y2**  Direct Estimation of Y2

**Y3**  Direct Estimation of Y3

**X1**  Auxiliary variable of X1

**X2**  Auxiliary variable of X2

**w1**  Known proportion of units in small areas of Y1

**w2**  Known proportion of units in small areas of Y2

**w3**  Known proportion of units in small areas of Y3

**v1**  Sampling Variance of Y1

**v12**  Sampling Covariance of Y1 and Y2

**v13**  Sampling Covariance of Y1 and Y3

**v2**  Sampling Variance of Y2

**v23**  Sampling Covariance of Y2 and Y3

**v3**  Sampling Variance of Y3

---

**msaedb**  

*EBLUPs under Multivariate Fay Herriot Model with Difference Benchmarking*

---

**Description**

This function produces EBLUPs, MSE, and aggregation of Multivariate SAE with Difference Benchmarking

**Usage**

```r
msaedb(
  formula,
  vardir,
  weight,
  samevar = FALSE,
  MAXITER = 100,
  PRECISION = 1e-04,
  data
)
```
Arguments

- **formula**: List of formula that describe the fitted model
- **vardir**: Sampling variances of direct estimations, if it is included in data frame so it is the vector with the name of sampling variances. If it is not, it is a data frame of sampling variance in order: var1, cov12, ..., cov1r, var2, cov23, ..., cov2r, ..., cov(r-1)r, var(r)
- **weight**: Known proportion of units in small areas, where \( \sum_{d=1}^{D} W_{rd} = 1 \). \( d = 1 \ldots D \) is the number of small areas, and \( r = 1 \ldots R \) is the number of response variables
- **samevar**: Whether the variances of the data are same or not. Logical input with default FALSE
- **MAXITER**: Maximum number of iteration in Fisher-scoring algorithm with default 100
- **PRECISION**: Limit of Fisher-scoring convergence tolerance with default 1e-4
- **data**: The data frame

Value

This function returns a list of the following objects:

- **MSAE_Eblup**: A dataframe with the values of the EBLUPs estimators
- **MSE_Eblup**: A dataframe with the values of estimated mean square errors of EBLUPs estimators
- **randomEffect**: A dataframe with the values of the random effect estimators
- **Rmatrix**: A block diagonal matrix composed of sampling errors
- **fit**: A list containing the following objects:
  - **method**: The fitting method (this function is using "REML")
  - **convergence**: The convergence result of Fisher-scoring algorithm (Logical Value)
  - **iterations**: The number of Fisher-Scoring algorithm iterations
  - **estcoef**: A dataframe with the estimated model coefficient, standard error, t statistics, p-values of the significance of each coefficient
  - **refvar**: A dataframe with estimated random effect variances
  - **informationFisher**: A matrix of information fisher from Fisher-scoring algorithm
- **difference_benchmarking**: a list containing the following objects:
  - **Estimation**: A dataframe with the value of Benchmarked EBLUPs estimators
  - **Aggregation**: The aggregation of benchmarked EBLUPs estimators, EBLUPs estimators and direct estimations
  - **MSE_DB**: A dataframe with the values of estimated mean square errors of benchmarked EBLUPs estimators
  - **g.4a**: First component of g4 in difference benchmarking MSE estimation formula
  - **g.4b**: Second component of g4 in difference benchmarking MSE estimation formula
Examples

```r
# load dataset
data(datamsaeDB)

# Compute fitted model for Y1, Y2, and Y3
# Y1 ~ X1 + X2
# Y2 ~ X2
# Y3 ~ X1

## Using parameter 'data'
formula = list(f1 = Y1 ~ X1 + X2,
               f2 = Y2 ~ X2,
               f3 = Y3 ~ X1)
vardir = c("v1", "v12", "v13", "v2", "v23", "v3")
weight = c("w1", "w2", "w3")
msaeDB <- msaedb(formula, vardir, weight, data = datamsaeDB)

## Do not use parameter 'data'
formula = list(f1 = datamsaeDB$Y1 ~ datamsaeDB$X1 + datamsaeDB$X2,
               f2 = datamsaeDB$Y2 ~ datamsaeDB$X2,
               f3 = datamsaeDB$Y3 ~ datamsaeDB$X1)
vardir = datamsaeDB[, c("v1", "v12", "v13", "v2", "v23", "v3")]
weight = datamsaeDB[, c("w1", "w2", "w3")]
msaeDB_d <- msaedb(formula, vardir, weight)

msaeDB$MSAE_Eblup # to see EBLUP Estimators
msaeDB$MSE_Eblup # to see estimated MSE of EBLUP estimators
msaeDB$difference_benchmarking$Estimation # to see Benchmarked EBLUP Estimators
msaeDB$difference_benchmarking$MSE_DB # to see estimated MSE of Benchmarked EBLUP Estimators
msaeDB$difference_benchmarking$Aggregation # to see the aggregation of benchmarking.
```

Description

This function produces EBLUPs, MSE, and aggregation of Multivariate SAE with Difference Benchmarking for non-sampled area.

Usage

```r
msaedbns(
  formula,
  vardir,
  weight,
  cluster,
)```
nonsample, samevar = FALSE, MAXITER = 100, PRECISION = 1e-04, data
)

Arguments

formula    List of formula that describe the fitted model
vardir     Sampling variances of direct estimations included in data frame as the vector with the name of sampling variances in order: var1, cov12, ... cov1r, var2, cov23, ... cov2r, ..., cov(r-1)r
weight     Known proportion of units in small areas, where \( \sum_{d=1}^{D} W_{rd} = 1 \), \( d = 1 \ldots D \) is the number of small areas, and \( r = 1 \ldots R \) is the number of response variables
cluster    cluster information
nonsample  A column with logical values, TRUE if the area is non-sampled
samevar    Whether the variances of the data are same or not. Logical input with default FALSE
MAXITER    Maximum number of iteration in Fisher-scoring algorithm with default 100
PRECISION  Limit of Fisher-scoring convergence tolerance with default 1e-4
data        The data frame

Value

This function returns a list of the following objects:

MSAE_Eblup_sampled   A dataframe with the values of the EBLUPs estimators for sampled areas
MSAE_Eblup_all       A dataframe with the values of the EBLUPs estimators for all areas
MSE_Eblup_sampled    A dataframe with the values of estimated mean square errors of EBLUPs estimators for sampled areas
MSE_Eblup_all        A dataframe with the values of estimated mean square errors of EBLUPs estimators for all areas
randomEffect_sampled a dataframe with the values of the random effect estimators for sampled areas
randomEffect_all     a dataframe with the values of the random effect estimators for all areas
Rmatrix_sampled      a block diagonal matrix composed of sampling errors for sampled areas
fit                  A list containing the following objects:
  • method : The fitting method (this function is using "REML")
  • convergence : The convergence result of Fisher-scoring algorithm (Logical Value)
  • iterations : The number of Fisher-Scoring algorithm iterations
• estcoef: A dataframe with the estimated model coefficient, standard error, t statistics, p-values of the significance of each coefficient
• refvar: A dataframe with estimated random effect variances
• informationFisher: A matrix of information fisher from Fisher-scoring algorithm

difference_benchmarking

A list containing the following objects:

• Estimation_sampled: A dataframe with the values of benchmarked EBLUPs estimators for sampled areas
• Estimation_all: A dataframe with the values of benchmarked EBLUPs estimators for all areas
• Aggregation_sampled: The aggregation of benchmarked EBLUPs estimators, EBLUPs estimators and direct estimations for sampled areas
• Aggregation_all: The aggregation of benchmarked EBLUPs estimators, EBLUPs estimators and direct estimations for all areas
• MSE_DB_sampled: A dataframe with the values of estimated mean square errors of benchmarked EBLUPs estimators for sampled areas
• MSE_DB_all: A dataframe with the values of estimated mean square errors of benchmarked EBLUPs estimators for all areas
• g.4a: First component of g4 in difference benchmarking MSE estimation formula
• g.4b: Second component of g4 in difference benchmarking MSE estimation formula

Examples

```r
# Load dataset
data(datamsaeDBns)
# Note: Make sure your dataset does not contain NA Values
# you can set 0 in Direct estimations and vardir for non-sampled areas.

data(datamsaeDBns)

# Compute Fitted model for Y1, Y2, and Y3
# Y1 ~ X1 + X2
# Y2 ~ X1 + X2
# Y3 ~ X1 + X2

# Using parameter 'data'
formula = list(f1 = Y1~X1+X2,
              f2 = Y2~X1+X2,
              f3 = Y3~X1+X2)
vardir = c("v1","v12","v13","v2","v23","v3")
weight = c("w1","w2","w3")
cluster = c("clY1","clY2","clY3")
nonsample = "nonsample"
msaeDBns <- msaedbs(formula, vardir, weight, cluster, nonsample, data=datamsaeDBns)
```
**msaefh**

*EBLUPs under Multivariate Fay Herriot Model*

**Description**

This function produces EBLUPs, MSE of Multivariate SAE

**Usage**

```r
msaefh(
  formula,
  vardir,
  samevar = FALSE,
  MAXITER = 100,
  PRECISION = 1e-04,
  data
)
```

**Arguments**

- **formula**: List of formula that describe the fitted model
- **vardir**: Sampling variances of direct estimations, if it is included in data frame so it is the vector with the name of sampling variances. If it is not, it is a data frame of sampling variance in order: var1, cov12, ..., cov1r, var2, cov23, ..., cov2r, ..., cov(r-1)r, var(r)
- **samevar**: Whether the variances of the data are same or not. Logical input with default FALSE
- **MAXITER**: Maximum number of iteration in Fisher-scoring algorithm with default 100
- **PRECISION**: Limit of Fisher-scoring convergence tolerance with default 1e-4
- **data**: The data frame

**Value**

This function returns a list of the following objects:

- **MSAE_Eblup**: A dataframe with the values of the EBLUPs estimators
- **MSE_Eblup**: A dataframe with the values of estimated mean square errors of EBLUPs estimators
- **randomEffect**: A dataframe with the values of the random effect estimators
- **Rmatrix**: A block diagonal matrix composed of sampling errors
- **fit**: A list containing the following objects:
  - **method**: The fitting method (this function is using "REML")
  - **convergence**: The convergence result of Fisher-scoring algorithm (Logical Value)
  - **iterations**: The number of Fisher-Scoring algorithm iterations
- estcoef: A dataframe with the estimated model coefficient, standard error, t statistics, p-values of the significance of each coefficient
- refvar: A dataframe with estimated random effect variances
- informationFisher: A matrix of information fisher from Fisher-scoring algorithm

Examples

```r
#load dataset
data(datamsaeDB)

#Compute Fitted model for Y1, Y2, and Y3
#Y1 ~ X1 + X2
#Y2 ~ X2
#Y3 ~ X1

##Using parameter ‘data’
formula = list(f1 = Y1~X1+X2,
               f2 = Y2~X2,
               f3 = Y3~X1)
vardir = c("v1","v12","v13","v2","v23","v3")
msaeFH <- msaefh(formula, vardir, data=datamsaeDB)

##Do not use parameter ‘data’
formula = list(f1 = datamsaeDB$Y1~datamsaeDB$X1+datamsaeDB$X2,
               f2 = datamsaeDB$Y2~datamsaeDB$X2,
               f3 = datamsaeDB$Y3~datamsaeDB$X1)
vardir = datamsaeDB[,c("v1","v12","v13","v2","v23","v3")]
msaeFH_d <- msaefh(formula, vardir)

msaeFH$MSAE_Eblup       #to see EBLUP Estimators
msaeFH$MSE_Eblup        #to see estimated MSE of EBLUP estimators
```

**msaeFhns**

**EBLUPs under Multivariate Fay Herriot Model for non-sampled area**

**Description**

This function produces EBLUPs and MSE of Multivariate SAE with Difference Benchmarking for non-sampled area

**Usage**

```r
msaeFhns(
  formula,  
  vardir,   
  cluster,  
  nonsample,  
  samevar = FALSE,
)```
msaefhns

\[
\begin{align*}
\text{MAXITER} &= 100, \\
\text{PRECISION} &= 1e-04, \\
\text{data} \\
\end{align*}
\]

Arguments

- **formula**: List of formula that describe the fitted model
- **vardir**: Sampling variances of direct estimations included in data frame as the vector with the name of sampling variances in order: \text{var1, cov12, ..., cov1r, var2, cov23, ..., cov(r-1)r, var(r)}
- **cluster**: Cluster information
- **nonsample**: A column with logical values, \text{TRUE} if the area is non-sampled
- **samevar**: Whether the variances of the data are same or not. Logical input with default \text{FALSE}
- **MAXITER**: Maximum number of iteration in Fisher-scoring algorithm with default 100
- **PRECISION**: Limit of Fisher-scoring convergence tolerance with default 1e-4
- **data**: The data frame

Value

This function returns a list of the following objects:

- **MSAE_Eblup_sampled**: A dataframe with the values of the EBLUPs estimators for sampled areas
- **MSAE_Eblup_all**: A dataframe with the values of the EBLUPs estimators for all areas
- **MSE_Eblup_sampled**: A dataframe with the values of estimated mean square errors of EBLUPs estimators for sampled areas
- **MSE_Eblup_all**: A dataframe with the values of estimated mean square errors of EBLUPs estimators for all areas
- **randomEffect_sampled**: A dataframe with the values of the random effect estimators for sampled areas
- **randomEffect_all**: A dataframe with the values of the random effect estimators for all areas
- **Rmatrix_sampled**: A block diagonal matrix composed of sampling errors for sampled areas
- **fit**: A list containing the following objects:
  - **method**: The fitting method (this function is using "REML")
  - **convergence**: The convergence result of Fisher-scoring algorithm (Logical Value)
  - **iterations**: The number of Fisher-Scoring algorithm iterations
  - **estcoef**: A dataframe with the estimated model coefficient, standard error, t statistics, p-values of the significance of each coefficient
  - **refvar**: A dataframe with estimated random effect variances
  - **informationFisher**: A matrix of information fisher from Fisher-scoring algorithm
Examples

```r
# Load dataset
data(datamsaeDBns)
# Note: Make sure your dataset does not contain NA values
# you can set 0 in Direct estimations and vardir for non-sampled areas.

# Compute fitted model for Y1, Y2, and Y3
# Y1 ~ X1 + X2
# Y2 ~ X1 + X2
# Y3 ~ X1 + X2

# Using parameter 'data'
formula = list(f1 = Y1 ~ X1 + X2,
              f2 = Y2 ~ X1 + X2,
              f3 = Y3 ~ X1 + X2)
vardir = c("v1","v12","v13","v2","v23","v3")
c cluster = c("clY1","clY2","clY3")
nonsample = "nonsample"
msaeFHns <- msaefhns(formula,vardir, cluster, nonsample, data=datamsaeDBns)
```

### Description

This function produces EBLUPs, MSE, and aggregation of Univariate SAE with Difference Benchmarking.

### Usage

```r
saedb(formula, vardir, weight, samevar = FALSE, MAXITER = 100, PRECISION = 1e-04, data)
```

### Arguments

- **formula**: List of formula that describe the fitted model
- **vardir**: Sampling variances of direct estimations, if it is included in data frame so it is the vector with the name of sampling variances. If it is not, it is a data frame of sampling variance in order: `var1, cov12, .., cov1r, var2, cov23, .., cov2r, .., cov(r-1)(r), var(r)`
weight

Known proportion of units in small areas, where \( \sum_{d=1}^{D} W_{rd} = 1 \). \( d = 1 \ldots D \) is the number of small areas, and \( r = 1 \ldots R \) is the number of response variables.

samevar

Whether the variances of the data are same or not. Logical input with default FALSE.

MAXITER

Maximum number of iteration in Fisher-scoring algorithm with default 100.

PRECISION

Limit of Fisher-scoring convergence tolerance with default 1e-4.

data

The data frame.

Value

This function returns a list of the following objects:

- SAE_Eblup: A dataframe with the values of the EBLUPs estimators
- MSE_Eblup: A dataframe with the values of estimated mean square errors of EBLUPs estimators
- randomEffect: A dataframe with the values of the random effect estimators
- Rmatrix: A block diagonal matrix composed of sampling errors
- fit: A list containing the following objects:
  - method: The fitting method (this function is using "REML")
  - convergence: The convergence result of Fisher-scoring algorithm (Logical Value)
  - iterations: The number of Fisher-Scoring algorithm iterations
  - estcoef: A dataframe with the estimated model coefficient, standard error, t statistics, p-values of the significance of each coefficient
  - refvar: A dataframe with estimated random effect variances
  - informationFisher: A matrix of information fisher from Fisher-scoring algorithm
- difference_benchmarking: A list containing the following objects:
  - Estimation: A dataframe with the value of Benchmarked EBLUPs estimators
  - Aggregation: The aggregation of benchmarked EBLUPs estimators, EBLUPs estimators and direct estimations
  - MSE_DB: A dataframe with the values of estimated mean square errors of benchmarked EBLUPs estimators
  - g.4a: First component of g4 in difference benchmarking MSE estimation formula
  - g.4b: Second component of g4 in difference benchmarking MSE estimation formula

Examples

```r
# load dataset
data(datamsaeDB)

# Compute Fitted model for Y1, Y2, and Y3
# Y1 ~ X1 + X2
```
#Y2 - X2
#Y3 - X1

# Using parameter 'data'
formula = list(f1 = Y1~X1+X2,
              f2 = Y2~X2,
              f3 = Y3~X1)
vardir = c("v1","v12","v13","v2","v23","v3")

# Note: in real data for univariate SAE, if you do not have the values of covariances, # set covariance as zero in the dataframe

weight = c("w1","w2","w3")
saeDB <- saedb(formula, vardir, weight, data=datamsaeDB)

# to calculate only one response variable
saeDB1 <- saedb(formula=list(f1=Y1~X1+X2),vardir = "v1", weight="w1",data=datamsaeDB )

# Do not use parameter 'data'
formula = list(f1 = datamsaeDB$Y1~datamsaeDB$X1+datamsaeDB$X2,
              f2 = datamsaeDB$Y2~datamsaeDB$X2,
              f3 = datamsaeDB$Y3~datamsaeDB$X1)
vardir = datamsaeDB[,c("v1","v12","v13","v2","v23","v3")]

# Note: in real data for univariate SAE, if you do not have the values of covariances, # set covariance as zero in the dataframe

weight = datamsaeDB[,c("w1","w2","w3")]
saeDB_d <- saedb(formula, vardir, weight = weight)

saeDB$SAE_Eblup # to see EBLUP estimators
saeDB$MSE_Eblup # to see estimated MSE of EBLUP estimators
saeDB$difference_benchmarking$Estimation # to see Benchmarked EBLUP estimators
saeDB$difference_benchmarking$MSE_DB # to see estimated MSE of Benchmarked EBLUP estimators
saeDB$difference_benchmarking$Aggregation # to see the aggregation of, benchmarking

---

**Description**

This function produces EBLUPs, MSE, and aggregation of Univariate SAE with Difference Benchmarking for non-sampled area
Usage

saedbns(
    formula,
    vardir,
    weight,
    cluster,
    nonsample,
    samevar = FALSE,
    MAXITER = 100,
    PRECISION = 1e-04,
    data
)

Arguments

formula List of formula that describe the fitted model
vardir Sampling variances of direct estimations included in data frame as the vector
    with the name of sampling variances in order: var1,cov12,..,cov1r,var2,cov23,..,cov2r,..,cov(r-1)r
weight Known proportion of units in small areas, where \( \sum_{d=1}^{D} W_{rd} = 1 \) . \( d = 1 \ldots D \) is
    the number of small areas, and \( r = 1 \ldots R \) is the number of response variables
cluster cluster information
nonsample A column with logical values, TRUE if the area is non-sampled
samevar Whether the variances of the data are same or not. Logical input with default
    FALSE
MAXITER Maximum number of iteration in Fisher-scoring algorithm with default 100
PRECISION Limit of Fisher-scoring convergence tolerance with default 1e-4
data The data frame

Value

This function returns a list of the following objects:

- **SAE_Eblup_sampled**: A dataframe with the values of the EBLUPs estimators for sampled areas
- **SAE_Eblup_all**: A dataframe with the values of the EBLUPs estimators for all areas
- **MSE_Eblup_sampled**: A dataframe with the values of estimated mean square errors of EBLUPs esti-
    mators for sampled areas
- **MSE_Eblup_all**: A dataframe with the values of estimated mean square errors of EBLUPs esti-
    mators for all areas
- **randomEffect_sampled**: A dataframe with the values of the random effect estimators for sampled areas
- **randomEffect_all**: A dataframe with the values of the random effect estimators for all areas
- **Rmatrix_sampled**: A block diagonal matrix composed of sampling errors for sampled areas
fit A list containing the following objects:

- method: The fitting method (this function is using "REML")
- convergence: The convergence result of Fisher-scoring algorithm (Logical Value)
- iterations: The number of Fisher-Scoring algorithm iterations
- estcoef: A dataframe with the estimated model coefficient, standard error, t statistics, p-values of the significance of each coefficient
- refvar: A dataframe with estimated random effect variances
- informationFisher: A matrix of information fisher from Fisher-scoring algorithm

difference_benchmarking

  a list containing the following objects:

- Estimation_sampled: A dataframe with the values of benchmarked EBLUPs estimators for sampled areas
- Estimation_all: A dataframe with the values of benchmarked EBLUPs estimators for all areas
- Aggregation_sampled: The aggregation of benchmarked EBLUPs estimators, EBLUPs estimators and direct estimations for sampled areas
- Aggregation_all: The aggregation of benchmarked EBLUPs estimators, EBLUPs estimators and direct estimations for all areas
- MSE_DB_sampled: A dataframe with the values of estimated mean square errors of benchmarked EBLUPs estimators for sampled areas
- MSE_DB_all: A dataframe with the values of estimated mean square errors of benchmarked EBLUPs estimators for all areas
- g.4a: First component of g4 in difference benchmarking MSE estimation formula
- g.4b: Second component of g4 in difference benchmarking MSE estimation formula

Examples

```r
#load dataset
data(dataSaeDBns)
#Note: Make sure your dataset does not contain NA Values
#
Compute Fitted model for Y1, Y2, and Y3
#Y1 ~ X1 + X2
#Y2 ~ X1 + X2
#Y3 ~ X1 + X2

#Using parameter 'data'
formula = list(f1 = Y1~X1+X2,
              f2 = Y2~X1+X2,
              f3 = Y3~X1+X2)
vardir = c("v1","v12","v13","v2","v23","v3")
weight = c("w1","w2","w3")
cluster = c("clY1","clY2","clY3")
```
nonsample = "nonsample"
saeDBns <- saedbs(formula,vardir, weight,cluster, nonsample, data=datamsaeDBns)

#to calculate only one response variable
saeDB1 <- saedbs(formula=list(f1=Y1~X1+X2),vardir ="v1", weight="w1",
                 cluster = "clY1",nonsample = "nonsample",data=datamsaeDBns )

---

**saefh**  
*EBLUPs under Univariate Fay Herriot Model*

**Description**

This function produces EBLUPs, MSE of Univariate SAE

**Usage**

`saefh(formula, vardir, samevar = FALSE, MAXITER = 100, PRECISION = 1e-04, data)`

**Arguments**

- *formula* List of formula that describe the fitted model
- *vardir* Sampling variances of direct estimations, if it is included in data frame so it is the vector with the name of sampling variances, if it is not, it is a data frame of sampling variance in order: var1, cov12, .., cov1r, var2, cov23, .., cov2r, .., cov(r-1)(r), var(r)
- *samevar* Whether the variances of the data are same or not. Logical input with default FALSE
- *MAXITER* Maximum number of iteration in Fisher-scoring algorithm with default 100
- *PRECISION* Limit of Fisher-scoring convergence tolerance with default 1e-4
- *data* The data frame

**Value**

This function returns a list of the following objects:

- **SAE_Eblup** A dataframe with the values of the EBLUPs estimators
- **MSE_Eblup** A dataframe with the values of estimated mean square errors of EBLUPs estimators
- **randomEffect** A dataframe with the values of the random effect estimators
- **Rmatrix** A block diagonal matrix composed of sampling errors
- **fit** A list containing the following objects:
  - *method* : The fitting method (this function is using "REML")
  - *convergence* : The convergence result of Fisher-scoring algorithm (Logical Value)
  - *iterations* : The number of Fisher-Scoring algorithm iterations
• estcoef: A dataframe with the estimated model coefficient, standard error, t statistics, p-values of the significance of each coefficient
• refvar: A dataframe with estimated random effect variances
• informationFisher: A matrix of information fisher from Fisher-scoring algorithm

Examples

```r
#load dataset
data(datamsaeDB)

#Compute Fitted model for Y1, Y2, and Y3
#Y1 ~ X1 + X2
#Y2 ~ X2
#Y3 ~ X1

##Using parameter 'data'
formula = list(f1 = Y1~X1+X2,
              f2 = Y2~X2,
              f3 = Y3~X1)
vardir = c("v1","v12","v13","v2","v23","v3")
#Note: in real data for univariate SAE, if you does not have the values of covariances,
# set covariances as zero in the dataframe
saeFH <- saefh(formula, vardir, data=datamsaeDB)

#to calculate only one response variable
saeFH1 <- saefh(formula=list(f1=Y1~X1+X2),vardir ="v1",data=datamsaeDB )

##Do not use parameter 'data'
formula = list(f1 = datamsaeDB$Y1~datamsaeDB$X1+datamsaeDB$X2,
              f2 = datamsaeDB$Y2~datamsaeDB$X2,
              f3 = datamsaeDB$Y3~datamsaeDB$X1)
vardir = datamsaeDB[,c("v1","v12","v13","v2","v23","v3")]
#Note: in real data for univariate SAE, if you does not have the values of covariances,
# set covariances as zero in the dataframe
saeFH_d <- saefh(formula, vardir)

saeFH$SAE_Eblup #to see EBLUP Estimators
saeFH$MSE_Eblup  #to see estimated MSE of EBLUP estimators
```

---

**saefhns**

**EBLUPs under Univariate Fay Herriot Model for non-sampled area**

**Description**

This function produces EBLUPs, MSE, and aggregation of Univariate SAE for non-sampled area
Usage

saefhns(
  formula,
  vardir,
  cluster,
  nonsample,
  samevar = FALSE,
  MAXITER = 100,
  PRECISION = 1e-04,
  data
)

Arguments

  formula List of formula that describe the fitted model
  vardir Sampling variances of direct estimations included in data frame as the vector
            with the name of sampling variances in order: var1, cov12, .., cov1r, var2, cov23, .., cov2r, .., cov(r-1)
  cluster cluster information
  nonsample A column with logical values, TRUE if the area is non-sampled
  samevar Whether the variances of the data are same or not. Logical input with default FALSE
  MAXITER Maximum number of iteration in Fisher-scoring algorithm with default 100
  PRECISION Limit of Fisher-scoring convergence tolerance with default 1e-4
  data The data frame

Value

This function returns a list of the following objects:

- SAE_Eblup_sampled
  - A dataframe with the values of the EBLUPs estimators for sampled areas
- SAE_Eblup_all
  - A dataframe with the values of the EBLUPs estimators for all areas
- MSE_Eblup_sampled
  - A dataframe with the values of estimated mean square errors of EBLUPs estimators for sampled areas
- MSE_Eblup_all
  - A dataframe with the values of estimated mean square errors of EBLUPs estimators for all areas
- randomEffect_sampled
  - A dataframe with the values of the random effect estimators for sampled areas
- randomEffect_all
  - A dataframe with the values of the random effect estimators for all areas
- Rmatrix_sampled
  - A block diagonal matrix composed of sampling errors for sampled areas
- fit
  - A list containing the following objects:

  - method : The fitting method (this function is using "REML")
• convergence: The convergence result of Fisher-scoring algorithm (Logical Value)
• iterations: The number of Fisher-Scoring algorithm iterations
• estcoef: A dataframe with the estimated model coefficient, standard error, t statistics, p-values of the significance of each coefficient
• refvar: A dataframe with estimated random effect variances
• informationFisher: A matrix of information fisher from Fisher-scoring algorithm

Examples

```r
# load dataset
data(datamsaeDBns)
# Note: Make sure your dataset does not contain NA Values
# you can set 0 in Direct estiminations and vardir for non-sampled areas.

# Compute Fitted model for Y1, Y2, and Y3
# Y1 ~ X1 + X2
# Y2 ~ X1 + X2
# Y3 ~ X1 + X2

# Using parameter 'data'
formula = list(f1 = Y1~X1+X2,
              f2 = Y2~X1+X2,
              f3 = Y3~X1+X2)
vardir = c("v1","v12","v13","v2","v23","v3")
cluster = c("clY1","clY2","clY3")
nonsample = "nonsample"
saefHns <- saefhns(formula, vardir, cluster, nonsample, data=datamsaeDBns)

# to calculate only one response variable
saefHns1 <- saefhns(formula=list(f1=Y1~X1+X2), vardir = "v1", cluster = "clY1",
                     nonsample = "nonsample", data=datamsaeDBns)
```
Index

* datasets
  datamsaeDB, 2
datamsaeDBns, 3

datamsaeDB, 2
datamsaeDBns, 3

msaedb, 4
msaedbns, 6
msaefh, 9
msaefhns, 10

saedb, 12
saedbns, 14
saefh, 17
saefhns, 18