Package 'sae.prop'

August 7, 2022

Type Package

Title Small Area Estimation using Fay-Herriot Models with Additive Logistic Transformation

Version 0.1.1

Description Implements Additive Logistic Transformation (alr) for Small Area Estimation under Fay Herriot Model. Small Area Estimation is used to borrow strength from auxiliary variables to improve the effectiveness of a domain sample size. This package uses Empirical Best Linear Unbiased Prediction (EBLUP). The Additive Logistic Transformation (alr) are based on transformation by Aitchison J (1986). The covariance matrix for multivariate application is based on covariance matrix used by Esteban M, Lombardía M, López-Vizcaíno E, Morales D, and Pérez A <doi:10.1007/s11749-019-00688-w>. The non-sampled models are modified area-level models based on models proposed by Anisa R, Kurnia A, and Indahwati I <doi:10.9790/5728-10121519>, with univariate model using model-3, and multivariate model using model-1. The MSE are estimated using Parametric Bootstrap approach. For non-sampled cases, MSE are estimated using modified approach proposed by Haris F and Ubaidillah A <doi:10.4108/eai.2-8-2019.2290339>.

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.2.0

URL https://github.com/mrijalussholihin/sae.prop

BugReports https://github.com/mrijalussholihin/sae.prop/issues

Imports stats, utils, magic, MASS, corpcor, progress, fpc

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Author M. Rijalus Sholihin [aut, cre],
Cucu Sumarni [aut]

Maintainer M. Rijalus Sholihin <221810400@stis.ac.id>

Repository CRAN

Date/Publication 2022-08-07 10:40:02 UTC

R topics documented:

`datasaem` *Data generated based on Multivariate Fay Herriot Model with Additive Logistic Transformation*

Description

This data is generated based on multivariate Fay-Herriot model and then transformed by using inverse Additive Logistic Transformation (alr). The steps are as follows:

1. Set these following variables:
   - $q = 4$
   - $r_1 = r_2 = r_3 = 2, r = 6$
   - $\beta_1 = (\beta_{11}, \beta_{12})' = (1,1)', \beta_2 = (\beta_{21}, \beta_{22})' = (1,1)', \beta_3 = (\beta_{31}, \beta_{32})' = (1,1)'$
   - $\mu_{x1} = \mu_{x2} = \mu_{x3}$ and $\sigma_{x11} = 1, \sigma_{x22} = 3/2, \sigma_{x33} = 2$
   - for $k = 1, 2, \ldots, q-1$ and $d = 1, \ldots, D$, generate $X_d = diag(x_{d1}, x_{d2}, x_{d3})_{(q-1)\times r}$, where:
     - $x_{d1} = (x_{d11}, x_{d11})$
     - $x_{d1} = (x_{d21}, x_{d22})$
     - $x_{d1} = (x_{d31}, x_{d31})$
     - $x_{d11} = x_{d21} = x_{d31} = 1$
     - $U_{dk} \sim U(0,1)$
     - $x_{d12} = \mu_{x1} + \sigma_{x11}^{1/2} U_{d1}$
     - $x_{d22} = \mu_{x2} + \sigma_{x22}^{1/2} U_{d2}$
     - $x_{d32} = \mu_{x3} + \sigma_{x33}^{1/2} U_{d3}$
2. For random effects u, $u_d \sim N_{q-1}(0, V_{ud})$, where $\theta_1 = 1, \theta_2 = 3/2, \theta_3 = 2, \theta_4 = -1/2, \theta_5 = -1/2, \theta_6 = 0$
3. For sampling errors e, $e_d \sim N_{q-1}(0, V_{ed})$, where $c = -1/4$

4. The generated data is transformed using inverse alr transformation, so the data will be within the range of proportion.

Auxiliary variables X_1, X_2, X_3, direct estimation Y_1, Y_2, Y_3, and sampling variance-covariance $v_1, v_2, v_3, v_{12}, v_{13}, v_{23}$ are combined into a data frame called datasaem. For more details about the structure of covariance matrix, it is available in supplementary materials of Reference.

Usage

```
datasaem
```

Format

A data frame with 30 rows and 12 columns:

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

X3 Auxiliary variable of X3

v1 Sampling Variance of Y1

v2 Sampling Variance of Y2

v3 Sampling Variance of Y3

v12 Sampling Covariance of Y1 and Y2

v13 Sampling Covariance of Y1 and Y3

v23 Sampling Covariance of Y2 and Y3

Reference

Esteban, M. D., Lombardía, M. J., López-Vizcaíno, E., Morales, D., & Pérez, A. (2020). Small area estimation of proportions under area-level compositional mixed models. Test, 29(3), 793–818. https://doi.org/10.1007/s11749-019-00688-w.

`datasaem.ns` *Data generated based on Multivariate Fay Herriot Model with Additive Logistic Transformation with Non-Sampled Cases*

Description

This data is generated based on multivariate Fay-Herriot model and then transformed by using inverse Additive Logistic Transformation (alr). Then some domain would be edited to be non-sampled. The steps are as follows:

1. Set these following variables:
   - $q = 4$
   - $r_1 = r_2 = r_3 = 2, r = 6$
   - $\beta_1 = (\beta_{11}, \beta_{12})' = (1, 1)', \beta_2 = (\beta_{21}, \beta_{22})' = (1, 1)', \beta_3 = (\beta_{31}, \beta_{32})' = (1, 1)'$
   - $\mu_{x1} = \mu_{x2} = \mu_{x3}$ and $\sigma_{x11} = 1, \sigma_{x22} = 3/2, \sigma_{x33} = 2$
   - for $k = 1, 2, \ldots, q-1$ and $d = 1, \ldots, D$, generate $X_d = diag(x_{d1}, x_{d2}, x_{d3})_{(q-1)\times r}$, where:
     - $x_{d1} = (x_{d11}, x_{d11})$
     - $x_{d1} = (x_{d21}, x_{d22})$
     - $x_{d1} = (x_{d31}, x_{d31})$
     - $x_{d11} = x_{d21} = x_{d31} = 1$
     - $U_{dk} \sim U(0, 1)$
     - $x_{d12} = \mu_{x1} + \sigma_{x11}^{1/2} U_{d1}$
     - $x_{d22} = \mu_{x2} + \sigma_{x22}^{1/2} U_{d2}$
     - $x_{d32} = \mu_{x3} + \sigma_{x33}^{1/2} U_{d3}$
2. For random effects u, $u_d \sim N_{q-1}(0, V_{ud})$, where $\theta_1 = 1, \theta_2 = 3/2, \theta_3 = 2, \theta_4 = -1/2, \theta_5 = -1/2, \theta_6 = 0$
3. For sampling errors e, $e_d \sim N_{q-1}(0, V_{ed})$, where $c = -1/4$
4. The generated data is transformed using inverse alr transformation, so the data will be within the range of proportion.
5. Domain 3, 15, and 25 are set to be examples of non-sampled cases (0, 1, or NA).
6. $c1$, $c2$, and $c3$ are clusters performed using k-medoids algorithm with `pamk`.

Auxiliary variables X_1, X_2, X_3, direct estimation Y_1, Y_2, Y_3, sampling variance-covariance $v_1, v_2, v_3, v_{12}, v_{13}, v_{23}$, and cluster $c1, c2, c3$ are combined into a data frame called datasaem.ns. For more details about the structure of covariance matrix, it is available in supplementary materials of Reference.

Usage

```
datasaem.ns
```

Format

A data frame with 30 rows and 15 columns:

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

X3 Auxiliary variable of X3

v1 Sampling Variance of Y1

v2 Sampling Variance of Y2

v3 Sampling Variance of Y3

v12 Sampling Covariance of Y1 and Y2

v13 Sampling Covariance of Y1 and Y3

v23 Sampling Covariance of Y2 and Y3

c1 Cluster of Y1

c2 Cluster of Y2

c3 Cluster of Y3

Reference

Esteban, M. D., Lombardía, M. J., López-Vizcaíno, E., Morales, D., & Pérez, A. (2020). Small area estimation of proportions under area-level compositional mixed models. Test, 29(3), 793–818. https://doi.org/10.1007/s11749-019-00688-w.

`datasaeu` *Data generated based on Univariate Fay Herriot Model with Additive Logistic Transformation*

Description

This data is generated based on univariate Fay-Herriot model and then transformed by using inverse Additive Logistic Transformation (alr). The steps are as follows:

1. β are set to be $\beta_0 = \beta_1 = \beta_2 = 1$
2. Auxiliary variables are set as follows:
   - $x_1 \sim N(0, 1)$
   - $x_2 \sim N(0.5, 1)$
3. For random effects, $u \sim N(0, V_u)$, where $V_u = 1$.
4. For sampling errors $e \sim N(0, V_{ed})$, where V_{ed} is generated $V_{ed} \sim InvGamma(50, 0.5)$.
5. The generated data is transformed using inverse alr transformation, so the data will be within the range of proportion.

Auxiliary variables x_1, x_2, direct estimation y, and sampling variance $vardir$ are combined into a data frame called datasaeu.

Usage

```
datasaeu
```

Format

A data frame with 30 rows and 4 columns:

y Direct Estimation of y

x1 Auxiliary variable of x1

x2 Auxiliary variable of x2

vardir Sampling Variance of y

`datasaeu.ns` *Data generated based on Univariate Fay Herriot Model with Additive Logistic Transformation with Non-Sampled Cases*

Description

This data is generated based on univariate Fay-Herriot model and then transformed by using inverse Additive Logistic Transformation (alr). Then some domain would be edited to be non-sampled. The steps are as follows:

1. β are set to be $\beta_0 = \beta_1 = \beta_2 = 1$
2. Auxiliary variables are set as follows:
   - $x_1 \sim N(0, 1)$
   - $x_2 \sim N(0.5, 1)$
3. For random effects, $u \sim N(0, V_u)$, where $V_u = 1$.
4. For sampling errors $e \sim N(0, V_{ed})$, where V_{ed} is generated $V_{ed} \sim InvGamma(50, 0.5)$.
5. The generated data is transformed using inverse alr transformation, so the data will be within the range of proportion.
6. Domain 3, 15, and 25 are set to be examples of non-sampled cases (0, 1, or NA).
7. $cluster$ is cluster performed using k-medoids algorithm with `pamk`.

Auxiliary variables x_1, x_2, direct estimation y, and sampling variance $vardir$ are combined into a data frame called datasaeu.

Usage

```
datasaeu.ns
```

Format

A data frame with 30 rows and 5 columns:

y Direct Estimation of y

x1 Auxiliary variable of x1

x2 Auxiliary variable of x2

vardir Sampling Variance of y

cluster Cluster of y

`mseFH.mprop` *Parametric Bootstrap Mean Squared Error of EBLUPs based on a Multivariate Fay Herriot model with Additive Logistic Transformation*

Description

This function gives the MSE of transformed EBLUP and Empirical Best Predictor (EBP) based on a multivariate Fay-Herriot model with modified parametric bootstrap approach proposed by Gonzalez-Manteiga.

Usage

```
mseFH.mprop(
  formula,
  vardir,
  MAXITER = 100,
  PRECISION = 1e-04,
  L = 1000,
  B = 400,
  data
)
```

Arguments

`formula`	an object of class `formula` that describe the fitted model.
`vardir`	sampling variances of direct estimations. If data is defined, it is a vector containing names of sampling variance columns. If data is not defined, it should be a data frame of sampling variances of direct estimators. The order is $var1, var2, \ldots, var(q-1), cov12, \ldots, cov1(q-1), cov23, \ldots, cov(q-2)(q-1)$.
`MAXITER`	maximum number of iterations allowed in the Fisher-scoring algorithm, Default: `100`.
`PRECISION`	convergence tolerance limit for the Fisher-scoring algorithm, Default: `1e-4`.
`L`	number of Monte Carlo iterations in calculating Empirical Best Predictor (EBP), Default: `1000`.
`B`	number of Bootstrap iterations in calculating MSE, Default: `400`.
`data`	optional data frame containing the variables named in `formula` and `vardir`.

Value

The function returns a list with the following objects:

`est` a list containing the following objects:

- `PC` : data frame containing transformed EBLUP estimators using inverse alr for each category.
- `EBP` : data frame containing Empirical Best Predictor using Monte Carlo for each category.

`fit` a list containing the following objects (model is fitted using REML):

- `convergence` : 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` : data frame that contains the estimated model coefficients, standard errors, t-statistics, and p-values of each coefficient.
- `refvar` : estimated covariance matrix of random effects.

`components` a list containing the following objects:

- `random.effects` : data frame containing estimated random effect values of the fitted model for each category.
- `residuals` : data frame containing residuals of the fitted model for each category.

`mse` a list containing estimated MSE of the estimators.

- `PC` : estimated MSE of plugin (PC) estimators for each category.
- `EBP` : estimated MSE of EBP estimators for each category.

Examples

```
## Load dataset
data(datasaem)

## If data is defined
Fo = list(Y1 ~ X1,
          Y2 ~ X2,
          Y3 ~ X3)
vardir = c("v1", "v2", "v3", "v12", "v13", "v23")
MSE.data <- mseFH.mprop(Fo, vardir, data = datasaem, B = 10)

## If data is undefined
Fo = list(datasaem$Y1 ~ datasaem$X1,
          datasaem$Y2 ~ datasaem$X2,
          datasaem$Y3 ~ datasaem$X3)
vardir = datasaem[, c("v1", "v2", "v3", "v12", "v13", "v23")]
MSE <- mseFH.mprop(Fo, vardir, B = 10)

## See the estimators
MSE$mse

## NOTE:
## B = 10 is just for examples.
## Please choose a proper number for Bootstrap iterations in real calculation.
```

mseFH.ns.mprop *Parametric Bootstrap Mean Squared Error of EBLUPs based on a Multivariate Fay Herriot model with Additive Logistic Transformation for Non-Sampled Data*

Description

This function gives the MSE of transformed EBLUP based on a multivariate Fay-Herriot model. For sampled domains, MSE is estimated using modified parametric bootstrap approach proposed by Gonzalez-Manteiga. For non-sampled domains, MSE is estimated using modified approach by using average sampling variance of sampled domain in each cluster.

Usage

```
mseFH.ns.mprop(
  formula,
  vardir,
  MAXITER = 100,
  PRECISION = 1e-04,
  cluster = "auto",
  B = 400,
  data
)
```

Arguments

formula	an object of class formula that describe the fitted model.
vardir	sampling variances of direct estimations. If data is defined, it is a vector containing names of sampling variance columns. If data is not defined, it should be a data frame of sampling variances of direct estimators. The order is $var1, var2, \ldots, var(q-1), cov12, \ldots, cov1(q-1), cov23, \ldots, cov(q-2)(q-1)$.
MAXITER	maximum number of iterations allowed in the Fisher-scoring algorithm, Default: 100.
PRECISION	convergence tolerance limit for the Fisher-scoring algorithm, Default: 1e-4.
cluster	Default: "auto". If cluster = "auto", then the clustering will be performed by the function by finding optimal number of cluster. If cluster is a vector containing numbers of cluster for each category, then clustering will be performed based on the chosen number of cluster. If cluster is a data frame or matrix containing cluster information, then the vector will be used directly to find average of random effects. Clustering is performed with k-medoids algorithms using the function pamk. If "auto" is chosen, krange are set to 2:(nrow(data)-1).
B	number of Bootstrap iterations in calculating MSE, Default: 400.
data	optional data frame containing the variables named in formula and vardir.

Value

The function returns a list with the following objects:

`est` a data frame containing values of the estimators for each domains.

- `PC` : transformed EBLUP estimators using inverse alr for each categoory.
- `status` : status of corresponding domain, whether sampled or non-sampled.

`fit` a list containing the following objects (model is fitted using REML):

- `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 that contains the estimated model coefficients, standard errors, t-statistics, and p-values of each coefficient.
- `refvar` : estimated covariance matrix of random effects.
- `cluster` : cluster of each category.
- `cluster.information` : a list containing data frames with average random effects of sampled domain in each cluster.

`components` a list containing the following objects:

- `random.effects` : data frame containing estimated random effect values of the fitted model for each category and their status whether sampled or non-sampled.
- `residuals` : data frame containing residuals of the fitted model for each category and their status whether sampled or non-sampled.

`mse` data frame containing estimated MSE of the estimators.

- `PC` : estimated MSE of plugin (PC) estimators for each category.
- `status` : status of domain, whether sampled or non-sampled.

Examples

```
## Load dataset
data(datasaem.ns)

## If data is defined
Fo = list(Y1 ~ X1,
          Y2 ~ X2,
          Y3 ~ X3)
vardir = c("v1", "v2", "v3", "v12", "v13", "v23")
MSE.ns <- mseFH.ns.mprop(Fo, vardir, data = datasaem.ns, B = 10)

## If data is undefined (and option for cluster arguments)
Fo = list(datasaem.ns$Y1 ~ datasaem.ns$X1,
          datasaem.ns$Y2 ~ datasaem.ns$X2,
          datasaem.ns$Y3 ~ datasaem.ns$X3)
```

```
vardir = datasaem.ns[, c("v1", "v2", "v3", "v12", "v13", "v23")]

### "auto"
MSE.ns1 <- mseFH.ns.mprop(Fo, vardir, cluster = "auto", B = 10)

### number of clusters
MSE.ns2 <- mseFH.ns.mprop(Fo, vardir, cluster = c(3, 2, 2), B = 10)

### data frame or matrix containing cluster for each domain
MSE.ns3 <- mseFH.ns.mprop(Fo, vardir, cluster = datasaem.ns[, c("c1", "c2", "c3")], B = 10)

## See the estimators
MSE.ns$mse

## NOTE:
## B = 10 is just for examples.
## Please choose a proper number for Bootstrap iterations in real calculation.
```

`mseFH.ns.uprop` *Parametric Bootstrap Mean Squared Error of EBLUPs based on a Univariate Fay Herriot model with Additive Logistic Transformation for Non-Sampled Data*

Description

This function gives the MSE of transformed EBLUP based on a univariate Fay-Herriot model. For sampled domains, MSE is estimated using modified parametric bootstrap approach proposed by Butar & Lahiri. For non-sampled domains, MSE is estimated using modified approach proposed by Haris & Ubaidillah.

Usage

```
mseFH.ns.uprop(
  formula,
  vardir,
  MAXITER = 100,
  PRECISION = 1e-04,
  cluster = "auto",
  B = 1000,
  data
)
```

Arguments

`formula`	an object of class `formula` that describe the fitted model.
`vardir`	vector containing the sampling variances of direct estimators for each domain. The values must be sorted as the variables in `formula`.

MAXITER	maximum number of iterations allowed in the Fisher-scoring algorithm, Default: 100.
PRECISION	convergence tolerance limit for the Fisher-scoring algorithm, Default: 1e-4.
cluster	Default: "auto". If cluster = "auto", then the clustering will be performed by the function by finding optimal number of cluster. If cluster is a number, then clustering will be performed based on the chosen number of cluster. If cluster is a vector containing cluster information, then the vector will be used directly to find average of random effects. Clustering is performed with k-medoids algorithms using the function pamk. If "auto" is chosen, krange are set to 2:(nrow(data)-1).
B	number of Bootstrap iterations in calculating MSE, Default: 1000.
data	optional data frame containing the variables named in formula and vardir.

Value

The function returns a list with the following objects:

est — a data frame containing values of the estimators for each domains.

- PC : transformed EBLUP estimators using inverse alr.
- status : status of corresponding domain, whether sampled or non-sampled.
- cluster : cluster of corresponding domain.

fit — a list containing the following objects (model is fitted using REML):

- 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 that contains the estimated model coefficients, standard errors, t-statistics, and p-values of each coefficient.
- refvar : estimated random effects variance.
- cluster.information : a data frame containing average random effects of sampled domain in each cluster.

components — a data frame containing the following columns:

- random.effects : estimated random effect values of the fitted model.
- residuals : residuals of the fitted model.
- status : status of corresponding domain, whether sampled or non-sampled.

mse — a data frame containing estimated MSE of the estimators.

- PC : estimated MSE of plugin (PC) estimators.
- status : status of domain, whether sampled or non-sampled.

Examples

```
## Load dataset
data(datasaeu.ns)

## If data is defined
Fo = y ~ x1 + x2
vardir = "vardir"
MSE.ns <- mseFH.ns.uprop(Fo, vardir, data = datasaeu.ns)

## If data is undefined (and option for cluster arguments)
Fo = datasaeu.ns$y ~ datasaeu.ns$x1 + datasaeu.ns$x2
vardir = datasaeu.ns$vardir

### "auto"
MSE.ns1 <- mseFH.ns.uprop(Fo, vardir, cluster = "auto")

### number of clusters
MSE.ns2 <- mseFH.ns.uprop(Fo, vardir, cluster = 2)

### vector containing cluster for each domain
MSE.ns3 <- mseFH.ns.uprop(Fo, vardir, cluster = datasaeu.ns$cluster)

## See the estimators
MSE.ns$mse
```

`mseFH.uprop` *Parametric Bootstrap Mean Squared Error of EBLUPs based on a Univariate Fay Herriot model with Additive Logistic Transformation*

Description

This function gives the MSE of transformed EBLUP and Empirical Best Predictor (EBP) based on a univariate Fay-Herriot model with modified parametric bootstrap approach proposed by Butar & Lahiri.

Usage

```
mseFH.uprop(
  formula,
  vardir,
  MAXITER = 100,
  PRECISION = 1e-04,
  L = 1000,
  B = 1000,
  data
)
```

Arguments

formula	an object of class formula that describe the fitted model.
vardir	vector containing the sampling variances of direct estimators for each domain. The values must be sorted as the variables in formula.
MAXITER	maximum number of iterations allowed in the Fisher-scoring algorithm, Default: 100.
PRECISION	convergence tolerance limit for the Fisher-scoring algorithm, Default: 1e-4.
L	number of Monte Carlo iterations in calculating Empirical Best Predictor (EBP), Default: 1000.
B	number of Bootstrap iterations in calculating MSE, Default: 1000.
data	optional data frame containing the variables named in formula and vardir.

Value

The function returns a list with the following objects:

est — a data frame containing values of the estimators for each domains.

- PC : transformed EBLUP estimators using inverse alr.
- EBP : Empirical Best Predictor using Monte Carlo.

fit — a list containing the following objects (model is fitted using REML):

- 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 that contains the estimated model coefficients, standard errors, t-statistics, and p-values of each coefficient.
- refvar : estimated random effects variance.

components — a data frame containing the following columns:

- random.effects : estimated random effect values of the fitted model.
- residuals : residuals of the fitted model.

mse — a data frame containing estimated MSE of the estimators.

- PC : estimated MSE of plugin (PC) estimators.
- EBP : estimated MSE of EBP estimators.

Examples

```
## Load dataset
data(datasaeu)

## If data is defined
Fo = y ~ x1 + x2
vardir = "vardir"
MSE.data <- mseFH.uprop(Fo, vardir, data = datasaeu)

## If data is undefined
Fo = datasaeu$y ~ datasaeu$x1 + datasaeu$x2
vardir = datasaeu$vardir
MSE <- mseFH.uprop(Fo, vardir)

## See the estimators
MSE$mse
```

`sae.prop` *sae.prop : Small Area Estimation for Proportion using Fay Herriot Models with Additive Logistic Transformation*

Description

Implements Additive Logistic Transformation (alr) for Small Area Estimation under Fay Herriot Model. Small Area Estimation is used to borrow strength from auxiliary variables to improve the effectiveness of a domain sample size. This package uses Empirical Best Linear Unbiased Prediction (EBLUP) estimator. The Additive Logistic Transformation (alr) are based on transformation by Aitchison J (1986). The covariance matrix for multivariate application is base on covariance matrix used by Esteban M, Lombardía M, López-Vizcaíno E, Morales D, and Pérez A <doi:10.1007/s11749-019-00688-w>. The non-sampled models are modified area-level models based on models proposed by Anisa R, Kurnia A, and Indahwati I <doi:10.9790/5728-10121519>, with univariate model using model-3, and multivariate model using model-1. The MSE are estimated using Parametric Bootstrap approach. For non-sampled cases, MSE are estimated using modified approach proposed by Haris F and Ubaidillah A <doi:10.4108/eai.2-8-2019.2290339>.

Author(s)

M. Rijalus Sholihin, Cucu Sumarni

Maintainer: M. Rijalus Sholihin <221810400@stis.ac.id>

Functions

`saeFH.uprop` EBLUPs based on a Univariate Fay Herriot model with Additive Logistic Transformation

`mseFH.uprop` Parametric Bootstrap Mean Squared Error of EBLUPs based on a Univariate Fay Herriot model with Additive Logistic Transformation

`saeFH.ns.uprop` EBLUPs based on a Univariate Fay Herriot model with Additive Logistic Transformation for Non-Sampled Data

`mseFH.ns.uprop` Parametric Bootstrap Mean Squared Error of EBLUPs based on a Univariate Fay Herriot model with Additive Logistic Transformation for Non-Sampled Data

`saeFH.mprop` EBLUPs based on a Multivariate Fay Herriot model with Additive Logistic Transformation

`mseFH.mprop` Parametric Bootstrap Mean Squared Error of EBLUPs based on a Multivariate Fay Herriot model with Additive Logistic Transformation

`saeFH.ns.mprop` EBLUPs based on a Multivariate Fay Herriot model with Additive Logistic Transformation for Non-Sampled Data

`mseFH.ns.mprop` Parametric Bootstrap Mean Squared Error of EBLUPs based on a Multivariate Fay Herriot model with Additive Logistic Transformation for Non-Sampled Data

Reference

- Rao, J.N.K & Molina. (2015). Small Area Estimation 2nd Edition. New York: John Wiley and Sons, Inc.
- Aitchison, J. (1986). The Statistical Analysis of Compositional Data. Springer Netherlands.
- Esteban, M. D., Lombardía, M. J., López-Vizcaíno, E., Morales, D., & Pérez, A. (2020). Small area estimation of proportions under area-level compositional mixed models. Test, 29(3), 793–818. https://doi.org/10.1007/s11749-019-00688-w.
- Anisa, R., Kurnia, A., & Indahwati, I. (2014). Cluster Information of Non-Sampled Area In Small Area Estimation. IOSR Journal of Mathematics, 10(1), 15–19. https://doi.org/10.9790/5728-10121519.
- Haris, F., & Ubaidillah, A. (2020, January 21). Mean Square Error of Non-Sampled Area in Small Area Estimation. https://doi.org/10.4108/eai.2-8-2019.2290339.

`saeFH.mprop` *EBLUPs based on a Multivariate Fay Herriot model with Additive Logistic Transformation*

Description

This function gives the transformed EBLUP and Empirical Best Predictor (EBP) based on a multivariate Fay-Herriot model. This function is used for multinomial compositional data. If data has P as proportion and total of q categories $(P_1 + P_2 + \ldots + P_q = 1)$, then function should be used to estimate $P_1, P_2, \ldots, P_{q-1}$.

Usage

```
saeFH.mprop(formula, vardir, MAXITER = 100, PRECISION = 1e-04, L = 1000, data)
```

Arguments

formula	an object of class formula that describe the fitted model.
vardir	sampling variances of direct estimations. If data is defined, it is a vector containing names of sampling variance columns. If data is not defined, it should be a data frame of sampling variances of direct estimators. The order is $var1, var2, \ldots, var(q-1), cov12, \ldots, cov1(q-1), cov23, \ldots, cov(q-2)(q-1)$.
MAXITER	maximum number of iterations allowed in the Fisher-scoring algorithm, Default: 100.
PRECISION	convergence tolerance limit for the Fisher-scoring algorithm, Default: 1e-4.
L	number of Monte Carlo iterations in calculating Empirical Best Predictor (EBP), Default: 1000.
data	optional data frame containing the variables named in formula and vardir.

Value

The function returns a list with the following objects:

est — a list containing data frame of the estimators for each domains.

- PC : transformed EBLUP estimators using inverse alr for each category.
- EBP : Empirical Best Predictor using Monte Carlo for each category.

fit — a list containing the following objects (model is fitted using REML):

- 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 that contains the estimated model coefficients, standard errors, t-statistics, and p-values of each coefficient.
- refvar : estimated covariance matrix of random effects.

components — a list containing the following objects:

- random.effects : data frame containing estimated random effect values of the fitted model for each category.
- residuals : data frame containing residuals of the fitted model for each category.

Examples

```
## Load dataset
data(datasaem)

## If data is defined
Fo = list(Y1 ~ X1,
          Y2 ~ X2,
          Y3 ~ X3)
vardir = c("v1", "v2", "v3", "v12", "v13", "v23")
```

```
model.data <- saeFH.mprop(Fo, vardir, data = datasaem)

Fo = list(datasaem$Y1 ~ datasaem$X1,
          datasaem$Y2 ~ datasaem$X2,
          datasaem$Y3 ~ datasaem$X3)
vardir = datasaem[, c("v1", "v2", "v3", "v12", "v13", "v23")]
model <- saeFH.mprop(Fo, vardir)

## See the estimators
model$est
```

`saeFH.ns.mprop` *EBLUPs based on a Multivariate Fay Herriot model with Additive Logistic Transformation for Non-Sampled Data*

Description

This function gives the transformed EBLUP based on a multivariate Fay-Herriot model. Random effects for sampled domains are from the fitted model and random effects for non-sampled domains are from cluster information. This function is used for multinomial compositional data. If data has P as proportion and total of q categories $(P_1 + P_2 + \ldots + P_q = 1)$, then function should be used to estimate $P_1, P_2, \ldots, P_{q-1}$.

Usage

```
saeFH.ns.mprop(
  formula,
  vardir,
  MAXITER = 100,
  PRECISION = 1e-04,
  cluster = "auto",
  data
)
```

Arguments

`formula`	an object of class `formula` that describe the fitted model.
`vardir`	sampling variances of direct estimations. If data is defined, it is a vector containing names of sampling variance columns. If data is not defined, it should be a data frame of sampling variances of direct estimators. The order is $var1, var2, \ldots, var(q-1), cov12, \ldots, cov1(q-1), cov23, \ldots, cov(q-2)(q-1)$.
`MAXITER`	maximum number of iterations allowed in the Fisher-scoring algorithm, Default: `100`.
`PRECISION`	convergence tolerance limit for the Fisher-scoring algorithm, Default: `1e-4`.

`cluster` Default: `"auto"`. If `cluster = "auto"`, then the clustering will be performed by the function by finding optimal number of cluster. If cluster is a vector containing numbers of cluster for each category, then clustering will be performed based on the chosen number of cluster. If cluster is a data frame or matrix containing cluster information, then the vector will be used directly to find average of random effects. Clustering is performed with k-medoids algorithms using the function `pamk`. If `"auto"` is chosen, `krange` are set to `2:(nrow(data)-1)`.

`data` optional data frame containing the variables named in `formula` and `vardir`.

Value

The function returns a list with the following objects:

`est` a data frame containing values of the estimators for each domains.

- `PC` : transformed EBLUP estimators using inverse alr for each categoory.
- `status` : status of corresponding domain, whether sampled or non-sampled.

`fit` a list containing the following objects (model is fitted using REML):

- `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 that contains the estimated model coefficients, standard errors, t-statistics, and p-values of each coefficient.
- `refvar` : estimated covariance matrix of random effects.
- `cluster` : cluster of each category.
- `cluster.information` : a list containing data frames with average random effects of sampled domain in each cluster.

`components` a list containing the following objects:

- `random.effects` : data frame containing estimated random effect values of the fitted model for each category and their status whether sampled or non-sampled.
- `residuals` : data frame containing residuals of the fitted model for each category and their status whether sampled or non-sampled.

Examples

```
## Load dataset
data(datasaem.ns)

## If data is defined
Fo = list(Y1 ~ X1,
          Y2 ~ X2,
          Y3 ~ X3)
vardir = c("v1", "v2", "v3", "v12", "v13", "v23")
model.ns <- saeFH.ns.mprop(Fo, vardir, data = datasaem.ns)
```

```
## If data is undefined (and option for cluster arguments)
Fo = list(datasaem.ns$Y1 ~ datasaem.ns$X1,
          datasaem.ns$Y2 ~ datasaem.ns$X2,
          datasaem.ns$Y3 ~ datasaem.ns$X3)
vardir = datasaem.ns[, c("v1", "v2", "v3", "v12", "v13", "v23")]

### "auto"
model.ns1 <- saeFH.ns.mprop(Fo, vardir, cluster = "auto")

### number of clusters
model.ns2 <- saeFH.ns.mprop(Fo, vardir, cluster = c(3, 2, 2))

### data frame or matrix containing cluster for each domain
model.ns3 <- saeFH.ns.mprop(Fo, vardir, cluster = datasaem.ns[, c("c1", "c2", "c3")])

## See the estimators
model.ns$est
```

`saeFH.ns.uprop` *EBLUPs based on a Univariate Fay Herriot model with Additive Logistic Transformation for Non-Sampled Data*

Description

This function gives the transformed EBLUP based on a univariate Fay-Herriot model. Random effects for sampled domains are from the fitted model and random effects for non-sampled domains are from cluster information.

Usage

```
saeFH.ns.uprop(
  formula,
  vardir,
  MAXITER = 100,
  PRECISION = 1e-04,
  cluster = "auto",
  data
)
```

Arguments

`formula`	an object of class `formula` that describe the fitted model.
`vardir`	vector containing the sampling variances of direct estimators for each domain. The values must be sorted as the variables in `formula`.
`MAXITER`	maximum number of iterations allowed in the Fisher-scoring algorithm, Default: `100`.
`PRECISION`	convergence tolerance limit for the Fisher-scoring algorithm, Default: `1e-4`.

`cluster` Default: `"auto"`. If `cluster = "auto"`, then the clustering will be performed by the function by finding optimal number of cluster. If cluster is a number, then clustering will be performed based on the chosen number of cluster. If cluster is a vector containing cluster information, then the vector will be used directly to find average of random effects. Clustering is performed with k-medoids algorithms using the function pamk. If `"auto"` is chosen, `krange` are set to `2:(nrow(data)-1)`.

`data` optional data frame containing the variables named in `formula` and `vardir`.

Value

The function returns a list with the following objects:

`est` a data frame containing values of the estimators for each domains.

- `PC` : transformed EBLUP estimators using inverse alr.
- `status` : status of corresponding domain, whether sampled or non-sampled.
- `cluster` : cluster of corresponding domain.

`fit` a list containing the following objects (model is fitted using REML):

- `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 that contains the estimated model coefficients, standard errors, t-statistics, and p-values of each coefficient.
- `refvar` : estimated random effects variance.
- `cluster.information` : a data frame containing average random effects of sampled domain in each cluster.

`components` a data frame containing the following columns:

- `random.effects` : estimated random effect values of the fitted model.
- `residuals` : residuals of the fitted model.
- `status` : status of corresponding domain, whether sampled or non-sampled.

Examples

```
## Load dataset
data(datasaeu.ns)

## If data is defined
Fo = y ~ x1 + x2
vardir = "vardir"
model.ns <- saeFH.ns.uprop(Fo, vardir, data = datasaeu.ns)

## If data is undefined (and option for cluster arguments)
Fo = datasaeu.ns$y ~ datasaeu.ns$x1 + datasaeu.ns$x2
```

```
vardir = datasaeu.ns$vardir

### "auto"
model.ns1 <- saeFH.ns.uprop(Fo, vardir, cluster = "auto")

### number of clusters
model.ns2 <- saeFH.ns.uprop(Fo, vardir, cluster = 2)

### vector containing cluster for each domain
model.ns3 <- saeFH.ns.uprop(Fo, vardir, cluster = datasaeu.ns$cluster)

## See the estimators
model.ns$est
```

`saeFH.uprop` *EBLUPs based on a Univariate Fay Herriot model with Additive Logistic Transformation*

Description

This function gives the transformed EBLUP and Empirical Best Predictor (EBP) based on a univariate Fay-Herriot model.

Usage

```
saeFH.uprop(formula, vardir, MAXITER = 100, PRECISION = 1e-04, L = 1000, data)
```

Arguments

`formula`	an object of class `formula` that describe the fitted model.
`vardir`	vector containing the sampling variances of direct estimators for each domain. The values must be sorted as the variables in `formula`.
`MAXITER`	maximum number of iterations allowed in the Fisher-scoring algorithm, Default: `100`.
`PRECISION`	convergence tolerance limit for the Fisher-scoring algorithm, Default: `1e-4`.
`L`	number of Monte Carlo iterations in calculating Empirical Best Predictor (EBP), Default: `1000`.
`data`	optional data frame containing the variables named in `formula` and `vardir`.

Value

The function returns a list with the following objects:

`est` a data frame containing values of the estimators for each domains.

- `PC` : transformed EBLUP estimators using inverse alr.

- EBP : Empirical Best Predictor using Monte Carlo.

`fit` a list containing the following objects (model is fitted using REML):

- `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 that contains the estimated model coefficients, standard errors, t-statistics, and p-values of each coefficient.
- `refvar` : estimated random effects variance.

`components` a data frame containing the following columns:

- `random.effects` : estimated random effect values of the fitted model.
- `residuals` : residuals of the fitted model.

Examples

```
## Load dataset
data(datasaeu)

## If data is defined
Fo = y ~ x1 + x2
vardir = "vardir"
model.data <- saeFH.uprop(Fo, vardir, data = datasaeu)

## If data is undefined
Fo = datasaeu$y ~ datasaeu$x1 + datasaeu$x2
vardir = datasaeu$vardir
model <- saeFH.uprop(Fo, vardir)

## See the estimators
model$est
```

Index