Package ‘saeMSPE’

October 21, 2022

Type Package

Title Compute MSPE Estimates for the Fay Herriot Model and Nested Error Regression Model

Version 1.2

Date 2022-10-19

Maintainer Peiwen Xiao <2569613200@qq.com>

Description We describe a new R package entitled 'saeMSPE' for the well-known Fay Herriot model and nested error regression model in small area estimation. Based on this package, it is possible to easily compute various common mean squared predictive error (MSPE) estimators, as well as several existing variance component predictors as a byproduct, for these two models.

License GPL (>= 2)

Depends R (>= 3.5.0), Matrix, smallarea

Imports Rcpp (>= 1.0.7)

LinkingTo Rcpp, RcppArmadillo

RoxygenNote 7.2.1

Encoding UTF-8

LazyData true

NeedsCompilation yes

Author Peiwen Xiao [aut, cre],
Xiaohui Liu [aut],
Yuzi Liu [aut],
Shaochu Liu [aut],
Jiming Jiang [ths]

Repository CRAN

Date/Publication 2022-10-21 12:12:37 UTC

R topics documented:

`saeMSPE-package` *Compute MSPE Estimates for the Fay Herriot Model and Nested Error Regression Model*

Description

We describe a new R package entitled 'saeMSPE' for the well-known Fay Herriot model and nested error regression model in small area estimation. Based on this package, it is possible to easily compute various common mean squared predictive error (MSPE) estimators, as well as several existing variance component predictors as a byproduct, for these two models.

Details

Package:	saeMSPE
Type:	Package
Version:	1.2
Date:	2022-10-19
License:	GPL (>=2)
Depends:	Matrix, smallarea

Author(s)

Peiwen Xiao, Xiaohui Liu, Yuzi Liu, Jiming Jiang, Shaochu Liu

Maintainer: Peiwen Xiao <2569613200@qq.com>

References

- V. Arora and P. Lahiri. On the superiority of the bayesian method over the blup in small area estimation problems. Statistica Sinica, 7(4):1053-1063, 1997.

- G. E. Battese, R. M. Harter, andW. A. Fuller. An error components model for prediction of county crop areas using survey and satellite data. Journal of the American Statistical Association, 83(401):28-36,1988.

- F. B. Butar and P. Lahiri. On measures of uncertainty of empirical bayes small area estimators. Journal of Statistical Planning and Inference, 112(1-2):63-76, 2003.

- G. S. Datta and P. Lahiri. A unified measure of uncertainty of estimated best linear unbiased predictors in small area estimation problems. Statistica Sinica, 10(2):613-627, 2000.

- G. S. Datta, J. N. K. Rao, and D. D. Smith. On measuring the variability of small area estimators under a basic area level model. Biometrika, 92(1):183-196, 2005.

- D. Eddelbuettel and C. Sanderson. Rcpparmadillo: Accelerating r with high-performance c++ linear algebra. Computational Statistics and Data Analysis, 71(March):1054-1063, 2014.

- D. Eddelbuettel, R. Francois, J. J. Allaire, K. Ushey, and Q. Kou. Rcpp: Seamless r and c++ integration. Journal of Statistical Software, 40(i08), 2011.

- R. E. Fay and R. A. Herriot. Estimates of income for small places: An application of james-stein procedures to census data. Journal of the American Statistical Association, 74(366a):269-277, 1979.

- W. González, M. J. Lombardía, I. Molina, D. Morales, and L. Santamaría. Bootstrap mean squared error of a small-area eblup. Journal of Statistical Computation and Simulation, 78(5):443-462, 2008.

- P. Hall and T. Maiti. On parametric bootstrap methods for small area prediction. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 2006a.

- P. Hall and T. Maiti. Nonparametric estimation of mean squared prediction error in nested error regression models. Annals of Statistics, 34(4):1733-1750, 2006b.

- C. R. Henderson. Best linear unbiased estimation and prediction under a selection model. Biometrics, 31(2):423-447, 1975.

- J. Jiang. Asymptotic Analysis of Mixed Effects Models: Theory, Applications, and Open Problems. 09 2017. ISBN 978-1-3151-1928-1. doi: 10.1201/9781315119281.

- J. Jiang and T. Nguyen. Linear and Generalized Linear Mixed Models and Their Applications. 01 2021. ISBN 978-1-0716-1281-1. doi: 10.1007/978-1-0716-1282-8.

- J. Jiang and M. Torabi. Sumca: simple; unified; monte carlo assisted approach to second order unbiased mean squared prediction error estimation. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 82(2):467-485, 2020

- J. Jiang and L. S. M.Wan. A unified jackknife theory for empirical best prediction with m estimation. Annals of Statistics, 30(6):1782-1810, 2002.

- J. Jiang, T. Nguyen, and J. S. Rao. Best predictive small area estimation. Journal of the American Statistical Association, 106(494):732-745, 2011.

- J. Jiang, P. Lahiri, and T. Nguyen. A unified monte carlo jackknife for small area estimation after model selection. Annals of Mathematiacal Sciences and Applications, 3(2):405-438, 2018.

- R. N. Kackar and D. A. Harville. Approximations for standard errors of estimators of fixed and random effects in mixed linear models. Journal of the American Statistical Association, 1984.

- X. Liu, H. Ma, and J. Jiang. That prasad-rao is robust: Estimation of mean squared prediction error of observed best predictor under potential model misspecification. Statistica Sinica, 2020.

- N. G. N. Prasad and J. N. K. Rao. The estimation of the mean squared error of small area estimators. Journal of the American Statistical Association, 85(409):163-171, 1990.

- M. H. Quenouille. Approximate tests of correlation in time series. Journal of the Royal Statistical Society. Series B (Methodological), 11(1):68-84, 1949.

- J. N. K. Rao and I. Molina. Small area estimation. Wiley, 2015.

- J. W. Tukey. Bias and confidence in not quite large samples. Annals of Mathematical Statistics, 29(2):614, 1958.

- Y. You and B. Chapman. Small area estimation using area level models and estimated sampling variances. Survey Methodology, 32(1):97-103, 2006.

`mspeFHdb` *Compute MSPE through double bootstrap method for Fay Herriot model*

Description

This function returns MSPE estimate with double bootstrap appoximation method for Fay Herriot model.

Usage

```
mspeFHdb(Y, X, D, K = 50, C = 50, method = 1)
```

Arguments

`Y`	(vector). It represents the response value for Fay Herriot model.
`X`	(matrix). Stands for the available auxiliary values.
`D`	(vector). It represents the knowing sampling variance for Fay Herriot model.
`K`	(integer). It represents the first bootstrap sample number. Default value is 50.
`C`	(integer). It represents the second bootstrap sample number. Default value is 50.
`method`	It represents the variance component estimation method. See "Details".

Details

This method was proposed by P. Hall and T. Maiti. Double bootstrap method uses boostrap tool twice for Fay Herriot model to avoid the unattractivitive bias correction: one is to estimate the estimator bias, the other is to correct for bias.

Default value for `method` is 1, `method = 1` represents the MOM method , `method = 2` and `method = 3` represents ML and REML method, respectively.

Value

A list with components:

- `MSPE` (vector) MSPE estimate based on double bootstrap method.
- `bhat` (vector) estimate of the unknown regression coefficients.
- `Ahat` (numeric) estimate of the variance component.

Author(s)

Peiwen Xiao, Xiaohui Liu, Yuzi Liu, Jiming Jiang, and Shaochu Liu

References

P. Hall and T. Maiti. On parametric bootstrap methods for small area prediction. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 2006.

Examples

```
X = matrix(runif(10 * 3), 10, 3)
X[,1] = rep(1, 10)
D = (1:10) / 10 + 0.5
Y = X %*% c(0.5,1,1.5) + rnorm(10, 0, sqrt(2)) + rnorm(10, 0, sqrt(D))
mspeFHdb(Y, X, D, K = 10, C = 10, 1)
```

`mspeFHjack` *Compute MSPE through Jackknife-based MSPE estimation method for Fay Herriot model*

Description

This function returns MSPE estimator with jackknife method for Fay Herriot model.

Usage

```
mspeFHjack(Y, X, D, method = 1)
```

Arguments

- `Y` (vector). It represents the response value for Fay Herriot model.
- `X` (matrix). It stands for the available auxiliary values.
- `D` (vector). Stands for the known sampling variances of each small area levels.
- `method` The variance component estimation method to be used. See "Details".

Details

This bias-corrected jackknife MSPE estimator was proposed by J. Jiang and L. S. M. Wan, it covers a fairly general class of mixed models which includes gLMM, mixed logistic model and some of the widely used mixed linear models as special cases.

Default value for `method` is 1, `method = 1` represents the MOM method , `method = 2` and `method = 3` represents ML and REML method, respectively.

Value

This function returns a list with components:

`MSPE`	(vector) MSPE estimates for Fay Herriot model.
`bhat`	(vector) Estimates of the unknown regression coefficients.
`Ahat`	(numeric) Estimates of the variance component.

Author(s)

Peiwen Xiao, Xiaohui Liu, Yuzi Liu, Jiming Jiang, and Shaochu Liu

References

M. H. Quenouille. Approximate tests of correlation in time series. *Journal of the Royal Statistical Society. Series B (Methodological)*, 11(1):68-84, 1949.

J. W. Tukey. Bias and confidence in not quite large samples. *Annals of Mathematical Statistics*, 29(2):614, 1958.

J. Jiang and L. S. M. Wan. A unified jackknife theory for empirical best prediction with m estimation. *Annals of Statistics*, 30(6):1782-1810, 2002.

Examples

```
X = matrix(runif(10 * 3), 10, 3)
X[,1] = rep(1, 10)
D = (1:10) / 10 + 0.5
Y = X %*% c(0.5,1,1.5) + rnorm(10, 0, sqrt(2)) + rnorm(10, 0, sqrt(D))
mspeFHjack(Y, X, D, method = 1)
```

`mspeFHlin` *Compute MSPE through linearization method for Fay Herriot model*

Description

This function returns MSPE estimator with linearization method for Fay Herriot model. These include the seminal Prasad-Rao method and its generalizations by Datta-Lahiri, Datta-Rao-Smith and Liu et.al. All these methods are developed for general linear mixed effects models.

Usage

```
mspeFHlin(Y, X, D, method = "PR", var.method = "default")

mspeFHPR(Y, X, D, var.method = "default")

mspeFHDL(Y, X, D, var.method = "default")

mspeFHDRS(Y, X, D, var.method = "default")

mspeFHMPR(Y, X, D, var.method = "default")
```

Arguments

`Y`	(vector). It represents the response value for Fay Herriot model.
`X`	(matrix). It stands for the available auxiliary values.
`D`	(vector). Stands for the known sampling variances of each small area levels.
`method`	The MSPE estimation method to be used. See "Details".
`var.method`	The variance component estimation method to be used. See "Details".

Details

Default `method` for `mspeFHlin` is "PR" ,proposed by N. G. N. Prasad and J. N. K. Rao, Prasad-Rao (PR) method uses Taylor series expansion to obtain a second-order approximation to the MSPE. Function `mspeFHlin` also provide the following methods:

Method "DL" proposed by Datta and Lahiri , It advanced PR method to cover the cases when the variance components are estimated by ML and REML estimator. Set `method = "DL"`.

Method "DRS" proposed by Datta and Smith, It focus on the second order unbiasedness appoximation when the variance component is replaced by Empirical Bayes estimator. Set `method = "DRS"`.

Method "MPR" is a modified version of "PR", It was proposed by Liu et al. It is a robust method that broaden the mean function from the linear form. Set `method = "MPR"`.

Default `var.method` and available variance component estimation method for each method is list as follows:

For `method = "PR"`, `var.method = "MOM"` is the only available variance component estimation method,

For `method = "DL"`, `var.method = "ML"` or `var.method = "REML"` is available,

For `method = "DRS"`, `var.method = "EB"` is the only available variance component estimation method,

For `method = "MPR"`, `var.method = "OBP"` is the only available variance component estimation method.

Value

This function returns a list with components:

`MSPE`	(vector) MSPE estimates for Fay Herriot model.
`bhat`	(vector) Estimates of the unknown regression coefficients.
`Ahat`	(numeric) Estimates of the variance component.

Author(s)

Peiwen Xiao, Xiaohui Liu, Yuzi Liu, Jiming Jiang, and Shaochu Liu

References

N. G. N. Prasad and J. N. K. Rao. The estimation of the mean squared error of small-area estimators. *Journal of the American Statistical Association*, 85(409):163-171, 1990.

G. S. Datta and P. Lahiri. A unified measure of uncertainty of estimated best linear unbiased predictors in small area estimation problems. *Statistica Sinica*, 10(2):613-627, 2000.

G. S. Datta and R. D. D. Smith. On measuring the variability of small area estimators under a basic area level model. *Biometrika*, 92(1):183-196, 2005.

X. Liu, H. Ma, and J. Jiang. That prasad-rao is robust: Estimation of mean squared prediction error of observed best predictor under potential model misspecification. *Statistica Sinica*, 2020.

Examples

```
X = matrix(runif(10 * 3), 10, 3)
X[,1] = rep(1, 10)
D = (1:10) / 10 + 0.5
Y = X %*% c(0.5,1,1.5) + rnorm(10, 0, sqrt(2)) + rnorm(10, 0, sqrt(D))
mspeFHlin(Y,X,D,method = "PR", var.method = "default")
```

---

mspeFHpb — *Compute MSPE through parameter bootstrap method for Fay Herriot model*

Description

This function returns MSPE estimator with parameter bootstrap method for Fay Herriot model.

Usage

```
mspeFHpb(Y, X, D, K = 50, method = 4)
```

Arguments

`Y`	(vector). It represents the response value for Fay Herriot model.
`X`	(matrix). Stands for the available auxiliary values.
`D`	(vector). It represents the knowing sampling variance for Fay Herriot model.
`K`	(integer). It represents the bootstrap sample number. Default value is 50.
`method`	The variance component estimation method to be used. See "Details".

Details

This method was proposed by Peter Hall and T. Maiti. Parametric bootstrap (pb) method uses bootstrap-based method to measure the accuracy of the EB estimator. In this case, only EB estimator is available (`method = 4`).

Value

This function returns a list with components:

MSPE	(vector) MSPE estimates for Fay Herriot model.
bhat	(vector) Estimates of the unknown regression coefficients.
Ahat	(numeric) Estimates of the variance component.

Author(s)

Peiwen Xiao, Xiaohui Liu, Yuzi Liu, Jiming Jiang, and Shaochu Liu

References

F. B. Butar and P. Lahiri. On measures of uncertainty of empirical bayes small area estimators. *Journal of Statistical Planning and Inference*, 112(1-2):63-76, 2003.

N. G. N. Prasad and J. N. K. Rao. The estimation of the mean squared error of small-area estimators. *Journal of the American Statistical Association*, 85(409):163-171, 1990.

Peter Hall and T. Maiti. On parametric bootstrap methods for small area prediction. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 2006a.

H. T. Maiti and T. Maiti. Nonparametric estimation of mean squared prediction error in nested error regression models. *Annals of Statistics*, 34(4):1733-1750, 2006b.

Examples

```
X = matrix(runif(10 * 3), 10, 3)
X[,1] = rep(1, 10)
D = (1:10) / 10 + 0.5
Y = X %*% c(0.5,1,1.5) + rnorm(10, 0, sqrt(2)) + rnorm(10, 0, sqrt(D))
mspeFHpb(Y, X, D, K = 50, method = 4)
```

---

mspeFHsumca *Compute MSPE through Sumca method for Fay Herriot model*

Description

This function returns MSPE estimator with the combination of linearization and resampling appoximation method called "Sumca", for Fay Herriot model.

Usage

```
mspeFHsumca(Y, X, D, K = 50, method = 1)
```

Arguments

Y	(vector). It represents the response value for Fay Herriot model.
X	(matrix). Stands for the available auxiliary values.
D	(vector). It represents the knowing sampling variance for Fay Herriot model.
K	(integer). It represents the Monte-Carlo sample size for "Sumca". Default value is 50.
method	It represents the variance component estimation method. See "Details".

Details

This method was proposed by J. Jiang, P. Lahiri, and T. Nguyen, sumca method combines the advantages of linearization and resampling methods and obtains unified, positive, low-computation burden and second-order unbiased MSPE estimators.

Default value for `method` is 1, `method = 1` represents the MOM method , `method = 2` and `method = 3` represents ML and REML method, respectively.

Value

This function returns a list with components:

MSPE	(vector) MSPE estimates for Fay Herriot model.
bhat	(vector) Estimates of the unknown regression coefficients.
Ahat	(numeric) Estimates of the variance component.

Author(s)

Peiwen Xiao, Xiaohui Liu, Yuzi Liu, Jiming Jiang, and Shaochu Liu

References

J. Jiang and M. Torabi. Sumca: simple; unified; monte carlo assisted approach to second order unbiased mean squared prediction error estimation. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 82(2):467-485, 2020.

Examples

```
X = matrix(runif(10 * 3), 10, 3)
X[,1] = rep(1, 10)
D = (1:10) / 10 + 0.5
Y = X %*% c(0.5,1,1.5) + rnorm(10, 0, sqrt(2)) + rnorm(10, 0, sqrt(D))
mspeFHsumca(Y, X, D, K = 50, method = 1)
```

`mspeNERdb` *Compute MSPE through double bootstrap(DB) method for Nested error regression model*

Description

This function returns MSPE estimator with double bootstrap method for Nested error regression model.

Usage

```
mspeNERdb(ni, X, Y, Xmean, K = 50, C = 50, method = 1)
```

Arguments

`ni`	(vector). It represents the sample number for every small area.
`X`	(matrix). It represents the small area response.
`Y`	(vector). It represents the design matrix.
`Xmean`	(matrix). Stands for the population mean of auxiliary values.
`K`	(integer). It represents the first bootstrap sample number. Default value is 50.
`C`	(integer). It represents the second bootstrap sample number. Default value is 50.
`method`	The variance component estimation method to be used. See "Details".

Details

This method was proposed by P. Hall and T. Maiti. Double bootstrap method uses boostrap tool twice for NER model to avoid the unattractivitive bias correction: one is to estimate the estimator bias, the other is to correct for bias.

Default value for `method` is 1, `method = 1` represents the MOM method , `method = 2` and `method = 3` represents ML and REML method, respectively.

Value

This function returns a list with components:

`MSPE`	(vector) MSPE estimates for NER model.
`bhat`	(vector) Estimates of the unknown regression coefficients.
`sigvhat2`	(numeric) Estimates of the area-specific variance component.
`sigehat2`	(numeric) Estimates of the random error variance component.

Author(s)

Peiwen Xiao, Xiaohui Liu, Yuzi Liu, Jiming Jiang, and Shaochu Liu

References

F. B. Butar and P. Lahiri. On measures of uncertainty of empirical bayes small area estimators. *Journal of Statistical Planning and Inference*, 112(1-2):63-76, 2003.

N. G. N. Prasad and J. N. K. Rao. The estimation of the mean squared error of small-area estimators. *Journal of the American Statistical Association*, 85(409):163-171, 1990.

Peter Hall and T. Maiti. On parametric bootstrap methods for small area prediction. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 2006a.

H. T. Maiti and T. Maiti. Nonparametric estimation of mean squared prediction error in nested error regression models. *Annals of Statistics*, 34(4):1733-1750, 2006b.

Examples

```
### parameter setting
Ni = 1000; sigmaX = 1.5; m = 10
beta = c(0.5, 1)
sigma_v2 = 0.8; sigma_e2 = 1
ni = sample(seq(1,10), m,replace = TRUE); n = sum(ni)
p = length(beta)
### population function
pop.model = function(Ni, sigmaX, beta, sigma_v2, sigma_e2, m){
  x = rnorm(m * Ni, 1, sqrt(sigmaX)); v = rnorm(m, 0, sqrt(sigma_v2)); y = numeric(m * Ni)
  theta = numeric(m); kk = 1
  for(i in 1 : m){
    sumx = 0
    for(j in 1:Ni){
      sumx = sumx + x[kk]
      y[kk] = beta[1] + beta[2] * x[kk] + v[i] + rnorm(1, 0, sqrt(sigma_e2))
      kk = kk + 1
    }
    meanx = sumx/Ni
    theta[i] = beta[1] + beta[2] * meanx + v[i]
  }
  group = rep(seq(m), each = Ni)
  x = cbind(rep(1, m*Ni), x)
  data = cbind(x, y, group)
  return(list(data = data, theta = theta))
}
### sample function
sampleXY = function(Ni, ni, m, Population){
  Indx = c()
  for(i in 1:m){
    Indx = c(Indx, sample(c(((i - 1) * Ni + 1) : (i * Ni)), ni[i]))
  }
  Sample = Population[Indx, ]; Nonsample = Population[-Indx, ]
  return(list(Sample, Nonsample))
}
### data generation process
Population = pop.model(Ni, sigmaX, beta, sigma_v2, sigma_e2, m)$data
XY = sampleXY(Ni, ni, m, Population)[[1]]
X = XY[, 1:p]
```

```
Y = XY[, p+1]
Xmean = matrix(NA, m, p)
for(tt in 1: m){
  Xmean[tt, ] = colMeans(Population[which(Population[,p+2] == tt), 1:p])
}
### mspe result
mspeNERdb(ni, X, Y, Xmean, 10, 10, method = 1)
```

---

mspeNERjack — *Compute MSPE through Jackknife-based MSPE estimation method for Nested error regression model*

Description

This function returns MSPE estimator with Jackknife-based MSPE estimation method for Nested error regression model.

Usage

```
mspeNERjack(ni, X, Y, Xmean, method = 1)
```

Arguments

ni	(vector). It represents the sample number for every small area.
X	(matrix). Stands for the available auxiliary values.
Y	(vector). It represents the response value for Nested error regression model.
Xmean	(matrix). Stands for the population mean of auxiliary values.
method	The MSPE estimation method to be used. See "Details".

Details

This bias-corrected jackknife MSPE estimator was proposed by J. Jiang and L. S. M. Wan, it covers a fairly general class of mixed models which includes gLMM, mixed logistic model and some of the widely used mixed linear models as special cases.

Default value for `method` is 1, `method = 1` represents the MOM method , `method = 2` and `method = 3` represents ML and REML method, respectively.

Value

This function returns a list with components:

MSPE	(vector) MSPE estimates for NER model.
bhat	(vector) Estimates of the unknown regression coefficients.
sigvhat2	(numeric) Estimates of the area-specific variance component.
sigehat2	(numeric) Estimates of the random error variance component.

Author(s)

Peiwen Xiao, Xiaohui Liu, Yuzi Liu, Jiming Jiang, and Shaochu Liu

References

M. H. Quenouille. Approximate tests of correlation in time series. *Journal of the Royal Statistical Society. Series B (Methodological)*, 11(1):68-84, 1949.

J. W. Tukey. Bias and confidence in not quite large samples. *Annals of Mathematical Statistics*, 29(2):614, 1958.

J. Jiang and L. S. M. Wan. A unified jackknife theory for empirical best prediction with m estimation. *Annals of Statistics*, 30(6):1782-1810, 2002.

Examples

```
### parameter setting
Ni = 1000; sigmaX = 1.5; m = 5
beta = c(0.5, 1)
sigma_v2 = 0.8; sigma_e2 = 1
ni = sample(seq(1,10), m,replace = TRUE); n = sum(ni)
p = length(beta)
### population function
pop.model = function(Ni, sigmaX, beta, sigma_v2, sigma_e2, m){
  x = rnorm(m * Ni, 1, sqrt(sigmaX)); v = rnorm(m, 0, sqrt(sigma_v2)); y = numeric(m * Ni)
  theta = numeric(m); kk = 1
  for(i in 1 : m){
    sumx = 0
    for(j in 1:Ni){
      sumx = sumx + x[kk]
      y[kk] = beta[1] + beta[2] * x[kk] + v[i] + rnorm(1, 0, sqrt(sigma_e2))
      kk = kk + 1
    }
    meanx = sumx/Ni
    theta[i] = beta[1] + beta[2] * meanx + v[i]
  }
  group = rep(seq(m), each = Ni)
  x = cbind(rep(1, m*Ni), x)
  data = cbind(x, y, group)
  return(list(data = data, theta = theta))
}
### sample function
sampleXY = function(Ni, ni, m, Population){
  Indx = c()
  for(i in 1:m){
    Indx = c(Indx, sample(c(((i - 1) * Ni + 1) : (i * Ni)), ni[i]))
  }
  Sample = Population[Indx, ]; Nonsample = Population[-Indx, ]
  return(list(Sample, Nonsample))
}
### data generation process
Population = pop.model(Ni, sigmaX, beta, sigma_v2, sigma_e2, m)$data
XY = sampleXY(Ni, ni, m, Population)[[1]]
```

```
X = XY[, 1:p]
Y = XY[, p+1]
Xmean = matrix(NA, m, p)
for(tt in 1: m){
  Xmean[tt, ] = colMeans(Population[which(Population[,p+2] == tt), 1:p])
}
### mspe result
mspeNERjack(ni, X, Y, Xmean, method = 1)
```

`mspeNERlin` *Compute MSPE through linearization method for Nested error regression model*

Description

This function returns MSPE estimator with linearization method for Nested error regression model. These include the seminal Prasad-Rao method and its generalizations by Datta-Lahiri. All these methods are developed for general linear mixed effects models.

Usage

```
mspeNERlin(ni, X, Y, X.mean, method = "PR", var.method = "default")

mspeNERPR(ni, X, Y, X.mean, var.method = "default")

mspeNERDL(ni, X, Y, X.mean, var.method = "default")
```

Arguments

`ni`	(vector). It represents the sample number for every small area.
`X`	(matrix). Stands for the available auxiliary values.
`Y`	(vector). It represents the response value for Nested error regression model.
`X.mean`	(matrix). Stands for the population mean of auxiliary values.
`method`	The MSPE estimation method to be used. See "Details".
`var.method`	The variance component estimation method to be used. See "Details".

Details

Default `method` for `mspeNERlin` is "PR" ,proposed by N. G. N. Prasad and J. N. K. Rao, Prasad-Rao (PR) method uses Taylor series expansion to obtain a second-order approximation to the MSPE. Function `mspeNERlin` also provide the following method:

Method "DL" advanced PR method to cover the cases when the variance components are estimated by ML and REML estimator. Set `method = "DL"`.

For `method = "PR"`, `var.method = "MOM"` is the only available variance component estimation method,

For `method = "DL"`, `var.method = "ML"` or `var.method = "REML"` are available.

Value

This function returns a list with components:

- `MSPE` (vector) MSPE estimates for NER model.
- `bhat` (vector) Estimates of the unknown regression coefficients.
- `sigvhat2` (numeric) Estimates of the area-specific variance component.
- `sigehat2` (numeric) Estimates of the random error variance component.

Author(s)

Peiwen Xiao, Xiaohui Liu, Yuzi Liu, Jiming Jiang, and Shaochu Liu

References

N. G. N. Prasad and J. N. K. Rao. The estimation of the mean squared error of small-area estimators. *Journal of the American Statistical Association*, 85(409):163-171, 1990.

G. S. Datta and P. Lahiri. A unified measure of uncertainty of estimated best linear unbiased predictors in small area estimation problems. *Statistica Sinica*, 10(2):613-627, 2000.

Examples

```
### parameter setting
Ni = 1000; sigmaX = 1.5; K = 100; C = 50; m = 10
beta = c(0.5, 1)
sigma_v2 = 0.8; sigma_e2 = 1
ni = sample(seq(1,10), m,replace = TRUE); n = sum(ni)
p = length(beta)
### population function
pop.model = function(Ni, sigmaX, beta, sigma_v2, sigma_e2, m){
  x = rnorm(m * Ni, 1, sqrt(sigmaX)); v = rnorm(m, 0, sqrt(sigma_v2)); y = numeric(m * Ni)
  theta = numeric(m); kk = 1
  for(i in 1 : m){
    sumx = 0
    for(j in 1:Ni){
      sumx = sumx + x[kk]
      y[kk] = beta[1] + beta[2] * x[kk] + v[i] + rnorm(1, 0, sqrt(sigma_e2))
      kk = kk + 1
    }
    meanx = sumx/Ni
    theta[i] = beta[1] + beta[2] * meanx + v[i]
  }
  group = rep(seq(m), each = Ni)
  x = cbind(rep(1, m*Ni), x)
  data = cbind(x, y, group)
  return(list(data = data, theta = theta))
}
### sample function
sampleXY = function(Ni, ni, m, Population){
  Indx = c()
  for(i in 1:m){
```

```
    Indx = c(Indx, sample(c(((i - 1) * Ni + 1) : (i * Ni)), ni[i]))
  }
  Sample = Population[Indx, ]; Nonsample = Population[-Indx, ]
  return(list(Sample, Nonsample))
}
### data generation process
Population = pop.model(Ni, sigmaX, beta, sigma_v2, sigma_e2, m)$data
XY = sampleXY(Ni, ni, m, Population)[[1]]
X = XY[, 1:p]
Y = XY[, p+1]
Xmean = matrix(NA, m, p)
for(tt in 1: m){
  Xmean[tt, ] = colMeans(Population[which(Population[,p+2] == tt), 1:p])
}
### mspe result
mspeNERlin(ni, X, Y, Xmean, method = "PR", var.method = "default")
```

`mspeNERpb` *Compute MSPE through parameter bootstrap method for Nested error regression model*

Description

This function returns MSPE estimator with parameter bootstrap appoximation method for Nested error regression model

Usage

```
mspeNERpb(ni, X, Y, Xmean, K = 50, method = 4)
```

Arguments

`ni`	(vector). It represents the sample number for every small area.
`Y`	(vector). It represents the response value for Nested error regression model.
`X`	(matrix). Stands for the available auxiliary values.
`Xmean`	(matrix). Stands for the population mean of auxiliary values.
`K`	(integer). It represents the bootstrap sample number. Default value is 50.
`method`	The variance component estimation method to be used. See "Details".

Details

This method was proposed by Peter Hall and T. Maiti. Parametric bootstrap (pb) method uses bootstrap-based method to measure the accuracy of EB estimator. In this case, only EB estimator is available (`method = 4`).

Value

This function returns a list with components:

MSPE	(vector) MSPE estimates for NER model.
bhat	(vector) Estimates of the unknown regression coefficients.
sigvhat2	(numeric) Estimates of the area-specific variance component.
sigehat2	(numeric) Estimates of the random error variance component.

Author(s)

Peiwen Xiao, Xiaohui Liu, Yuzi Liu, Jiming Jiang, and Shaochu Liu

References

F. B. Butar and P. Lahiri. On measures of uncertainty of empirical bayes small area estimators. *Journal of Statistical Planning and Inference*, 112(1-2):63-76, 2003.

N. G. N. Prasad and J. N. K. Rao. The estimation of the mean squared error of small-area estimators. *Journal of the American Statistical Association*, 85(409):163-171, 1990.

Peter Hall and T. Maiti. On parametric bootstrap methods for small area prediction. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 2006a.

H. T. Maiti and T. Maiti. Nonparametric estimation of mean squared prediction error in nested error regression models. *Annals of Statistics]*, 34(4):1733-1750, 2006b.

Examples

```
### parameter setting
Ni = 1000; sigmaX = 1.5; K = 50; C = 50; m = 10
beta = c(0.5, 1)
sigma_v2 = 0.8; sigma_e2 = 1
ni = sample(seq(1,10), m,replace = TRUE); n = sum(ni)
p = length(beta)
### population function
pop.model = function(Ni, sigmaX, beta, sigma_v2, sigma_e2, m){
  x = rnorm(m * Ni, 1, sqrt(sigmaX)); v = rnorm(m, 0, sqrt(sigma_v2)); y = numeric(m * Ni)
  theta = numeric(m); kk = 1
  for(i in 1 : m){
    sumx = 0
    for(j in 1:Ni){
      sumx = sumx + x[kk]
      y[kk] = beta[1] + beta[2] * x[kk] + v[i] + rnorm(1, 0, sqrt(sigma_e2))
      kk = kk + 1
    }
    meanx = sumx/Ni
    theta[i] = beta[1] + beta[2] * meanx + v[i]
  }
  group = rep(seq(m), each = Ni)
  x = cbind(rep(1, m*Ni), x)
  data = cbind(x, y, group)
  return(list(data = data, theta = theta))
```

```
}
### sample function
sampleXY = function(Ni, ni, m, Population){
  Indx = c()
  for(i in 1:m){
    Indx = c(Indx, sample(c(((i - 1) * Ni + 1) : (i * Ni)), ni[i]))
  }
  Sample = Population[Indx, ]; Nonsample = Population[-Indx, ]
  return(list(Sample, Nonsample))
}
### data generation process
Population = pop.model(Ni, sigmaX, beta, sigma_v2, sigma_e2, m)$data
XY = sampleXY(Ni, ni, m, Population)[[1]]
X = XY[, 1:p]
Y = XY[, p+1]
Xmean = matrix(NA, m, p)
for(tt in 1: m){
  Xmean[tt, ] = colMeans(Population[which(Population[,p+2] == tt), 1:p])
}
### mspe result
mspeNERpb(ni, X, Y, Xmean, 50)
```

`mspeNERsumca` *Compute MSPE through Sumca method for Nested error regression model*

Description

This function returns MSPE estimator with the combination of linearization and resampling appoximation method for Nested error regression model.

Usage

```
mspeNERsumca(ni, X, Y, Xmean, K = 50, method = 2)
```

Arguments

`ni`	(vector). It represents the sample number for every small area.
`X`	(matrix). Stands for the available auxiliary values.
`Y`	(vector). It represents the response value for Nested error regression model.
`Xmean`	(matrix). Stands for the population mean of auxiliary values.
`K`	(integer). It represents the Monte-Carlo sample size for "Sumca". Default value is 50.
`method`	The MSPE estimation method to be used. See "Details".

Details

This method was proposed by J. Jiang, P. Lahiri, and T. Nguyen, sumca method combines the advantages of linearization and resampling methods and obtains unified, positive, low-computation burden and second-order unbiased MSPE estimators.

Default value for `method` is 1, `method = 1` represents the MOM method , `method = 2` and `method = 3` represents ML and REML method, respectively.

Value

This function returns a list with components:

`MSPE`	(vector) MSPE estimates for NER model.
`bhat`	(vector) Estimates of the unknown regression coefficients.
`sigvhat2`	(numeric) Estimates of the area-specific variance component.
`sigehat2`	(numeric) Estimates of the random error variance component.

Author(s)

Peiwen Xiao, Xiaohui Liu, Yuzi Liu, Jiming Jiang, and Shaochu Liu

References

J. Jiang and M. Torabi. Sumca: simple; unified; monte carlo assisted approach to second order unbiased mean squared prediction error estimation. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 82(2):467-485, 2020.

Examples

```
### parameter setting
Ni = 1000; sigmaX = 1.5; m = 10
beta = c(0.5, 1)
sigma_v2 = 0.8; sigma_e2 = 1
ni = sample(seq(1,10), m,replace = TRUE); n = sum(ni)
p = length(beta)
### population function
pop.model = function(Ni, sigmaX, beta, sigma_v2, sigma_e2, m){
 x = rnorm(m * Ni, 1, sqrt(sigmaX)); v = rnorm(m, 0, sqrt(sigma_v2)); y = numeric(m * Ni)
  theta = numeric(m); kk = 1
  for(i in 1 : m){
    sumx = 0
    for(j in 1:Ni){
      sumx = sumx + x[kk]
      y[kk] = beta[1] + beta[2] * x[kk] + v[i] + rnorm(1, 0, sqrt(sigma_e2))
      kk = kk + 1
    }
    meanx = sumx/Ni
    theta[i] = beta[1] + beta[2] * meanx + v[i]
  }
  group = rep(seq(m), each = Ni)
  x = cbind(rep(1, m*Ni), x)
```

```
  data = cbind(x, y, group)
  return(list(data = data, theta = theta))
}
### sample function
sampleXY = function(Ni, ni, m, Population){
  Indx = c()
  for(i in 1:m){
    Indx = c(Indx, sample(c(((i - 1) * Ni + 1) : (i * Ni)), ni[i]))
  }
  Sample = Population[Indx, ]; Nonsample = Population[-Indx, ]
  return(list(Sample, Nonsample))
}
### data generation process
Population = pop.model(Ni, sigmaX, beta, sigma_v2, sigma_e2, m)$data
XY = sampleXY(Ni, ni, m, Population)[[1]]
X = XY[, 1:p]
Y = XY[, p+1]
Xmean = matrix(NA, m, p)
for(tt in 1: m){
  Xmean[tt, ] = colMeans(Population[which(Population[,p+2] == tt), 1:p])
}
### mspe result
mspeNERsumca(ni, X, Y, Xmean, 50, method = 1)
```

---

varfh *Estimates of the variance component using several methods for Fay Herriot model.*

Description

This function returns the estimate of variance component with several existing method for Fay Herriot model. This function does not accept missing values.

Usage

```
varfh(Y, X, D, method)

varOBP(Y, X, D)
```

Arguments

`Y`	(vector). It represents the response value for Fay Herriot model.
`X`	(matrix). It stands for the available auxiliary values.
`D`	(vector). It represents the knowing sampling variance for Fay Herriot model.
`method`	Variance component estimation method. See "Details".

Details

Default value for `method` is 1, It represents the moment estimator, Also called ANOVA estimator, The available variance component estimation method are list as follows:

`method = 1` represents the moment (MOM) estimator, ;

`method = 2` represents the restricted maximum likelihood (REML) estimator;

`method = 3` represents the maximum likelihood (ML) estimator;

`method = 4` represents the empirical bayesian (EB) estimator;

Value

This function returns a list with components:

`bhat`	(vector) Estimates of the unknown regression coefficients.
`Ahat`	(numeric) Estimates of the variance component.

Author(s)

Peiwen Xiao, Xiaohui Liu, Yuzi Liu, Jiming Jiang, and Shaochu Liu

References

J. Jiang. Linear and Generalized Linear Mixed Models and Their Applications. 2007.

Examples

```
X = matrix(runif(10 * 3), 10, 3)
X[,1] = rep(1, 10)
D = (1:10) / 10 + 0.5
Y = X %*% c(0.5,1,1.5) + rnorm(10, 0, sqrt(2)) + rnorm(10, 0, sqrt(D))
varOBP(Y, X, D)
varfh(Y, X, D, 1)
```

`varner` *Estimates of the variance component using several methods for Nested error regression model.*

Description

This function returns the estimate of variance component with several existing method for Nested error regression model. This function does not accept missing values.

Usage

```
varner(ni, X, Y, method)
```

Arguments

- `ni` (vector). It represents the sample number for every small area.
- `X` (matrix). Stands for the available auxiliary values.
- `Y` (vector). It represents the response value for Nested error regression model.
- `method` The variance component estimation method to be used. See "Details".

Details

Default value for `method` is 1, It represents the moment estimator, Also called ANOVA estimator, The available variance component estimation method are list as follows:

`method` = 1 represents the MOM estimator;

`method` = 2 represents the restricted maximum likelihood (REML) estimator;

`method` = 3 represents the maximum likelihood (ML) estimator;

`method` = 4 represents the empirical bayesian (EB) estimator;

Value

This function returns a list with components:

- `bhat` (vector) Estimates of the unknown regression coefficients.
- `sigvhat2` (numeric) Estimates of the area-specific variance component.
- `sigehat2` (numeric) Estimates of the random error variance component.

Author(s)

Peiwen Xiao, Xiaohui Liu, Yuzi Liu, Jiming Jiang, and Shaochu Liu

References

J. Jiang. Linear and Generalized Linear Mixed Models and Their Applications. 2007.

Examples

```
### parameter setting
Ni = 1000; sigmaX = 1.5; m = 10
beta = c(0.5, 1)
sigma_v2 = 0.8; sigma_e2 = 1
ni = sample(seq(1,10), m,replace = TRUE); n = sum(ni)
p = length(beta)
### population function
pop.model = function(Ni, sigmaX, beta, sigma_v2, sigma_e2, m){
  x = rnorm(m * Ni, 1, sqrt(sigmaX)); v = rnorm(m, 0, sqrt(sigma_v2)); y = numeric(m * Ni)
  theta = numeric(m); kk = 1
  for(i in 1 : m){
    sumx = 0
    for(j in 1:Ni){
      sumx = sumx + x[kk]
      y[kk] = beta[1] + beta[2] * x[kk] + v[i] + rnorm(1, 0, sqrt(sigma_e2))
```

```
    kk = kk + 1
   }
   meanx = sumx/Ni
   theta[i] = beta[1] + beta[2] * meanx + v[i]
  }
  group = rep(seq(m), each = Ni)
  x = cbind(rep(1, m*Ni), x)
  data = cbind(x, y, group)
  return(list(data = data, theta = theta))
}
### sample function
sampleXY = function(Ni, ni, m, Population){
  Indx = c()
  for(i in 1:m){
   Indx = c(Indx, sample(c(((i - 1) * Ni + 1) : (i * Ni)), ni[i]))
  }
  Sample = Population[Indx, ]; Nonsample = Population[-Indx, ]
  return(list(Sample, Nonsample))
}
### data generation process
Population = pop.model(Ni, sigmaX, beta, sigma_v2, sigma_e2, m)$data
XY = sampleXY(Ni, ni, m, Population)[[1]]
X = XY[, 1:p]
Y = XY[, p+1]
### variance component estimate
varner(ni,X,Y,1)
```

`wheatarea` *Wheat area measurement and satellite data.*

Description

Wheat area data measured at the scene in the block of Yanzhou District, Jining City, Shandong Province. The data corresponding to each block comes from the ArcGIS platform. The whole dataset consists of a total number of 458 villages and 14750 wheat blocks.

Usage

```
data(wheatarea)
```

Format

A data frame with 14708 observations on the following 3 variables.

`pixel`**:** Pixel sizes of each wheat blocks.

`F_AREA`**:** Field inspection area of each wheat blocks.

`code`**:** Street code.

Source

- Liu Y, Qu W, Cui Z, Liu X, Xu W, Jiang j,. (2021). Estimation of Wheat Growing Area via Mixed Model Prediction Using Satellite Data. Journal of Applied Statistics and Management.

Index