

Package ‘saeHB.ZIB’

May 9, 2026

Type Package

Title Small Area Estimation using Hierarchical Bayesian under Zero Inflated Binomial Distribution

Version 0.1.1

Author Rizqina Rahmati, Azka Ubaidillah

Maintainer Rizqina Rahmati <221810583@stis.ac.id>

Description Provides function for area level of small area estimation using hierarchical Bayesian (HB) method with Zero-Inflated Binomial distribution for variables of interest. Some dataset produced by a data generation are also provided. The 'rjags' package is employed to obtain parameter estimates. Model-based estimators involves the HB estimators which include the mean and the variation of mean.

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.1.2

Imports stringr, coda, rjags, stats, grDevices, graphics

Suggests rmarkdown, knitr

VignetteBuilder knitr

Depends R (>= 2.10)

NeedsCompilation no

Repository CRAN

Date/Publication 2022-06-22 08:40:02 UTC

Contents

dataZIB	2
dataZIBns	3
ziBinomial	3
Index	6

dataZIB	<i>Sample Data for Small Area Estimation using Hierarchical Bayesian Method under Zero-Inflated Binomial Distribution</i>
---------	---

Description

Dataset to simulate Small Area Estimation using Hierarchical Bayesian Method under Zero-Inflated Binomial distribution

This data is generated by these following steps:

1. Generate sampling random area effect $u.Z$ and $u.nZ$ with $(u.Z \sim N(0, 1))$ and $(u.nZ \sim N(0, 1))$.
The auxiliary variables are generated by Uniform distribution with $(x1 \sim U(0, 1))$ and $(x2 \sim U(1, 5))$.
The coefficient parameters $\alpha_0, \alpha_1, \alpha_2, \beta_0, \beta_1, \beta_2$ are set as 0.
2. Calculate $\text{logit}(p) = \alpha_0 + \alpha_1 * x1 + \alpha_2 * x2 + u.Z$ and $\text{logit}(\pi) = \beta_0 + \beta_1 * x1 + \beta_2 * x2 + u.nZ$
3. Generate number of sample with $n.samp \sim U(10, 30)$
4. Generate $\text{delta} \sim \text{bernoulli}(p)$ and $y_{star} \sim \text{binomial}(s, \pi)$
5. calculate $y = \text{delta} * y_{star}$
6. Calculate variance of direct estimates (`vardir`) with $\text{var}(y) = (1-p) * s * p * (1-p * (1-p * s))$
7. Auxiliary variables $x1, x2$, direct estimation (y), `vardir`, and s are combined in a dataframe called `dataZIB`

Usage

```
data(dataZIB)
```

Format

A data frame with 64 observations on the following 4 variables:

y Direct Estimation of y

X1 Auxiliary variable of x1

X2 Auxiliary variable of x2

vardir sampling variance of y

s number of sample

dataZIBns	<i>Sample Data for Small Area Estimation using Hierarchical Bayesian Method under Zero-Inflated Binomial Distribution</i>
-----------	---

Description

Dataset to simulate Small Area Estimation using Hierarchical Bayesian Method under Zero-Inflated Binomial distribution with non-sampled areas

This data contains NA values that indicates no sampled at one or more small areas. It uses the dataZIB.ns with the direct estimates and the related variances in 3 small areas are missing.

Usage

```
data(dataZIBns)
```

Format

A data frame with 30 rows and 4 variables :

y Direct Estimation of y

X1 Auxiliary variable of x1

X2 Auxiliary variable of x2

vardir sampling variance of y

s number of sample

ziBinomial	<i>Small Area Estimation using Hierarchical Bayesian under Zero Inflated Binomial Distribution</i>
------------	--

Description

This function is implemented to variable of interest (y) that assumed to be a Zero Inflated Binomial Distribution. The range of data is ($0 < y < \infty$). This model can be used to handle overdispersion caused by excess zero in data.

Usage

```
ziBinomial(  
  formula,  
  n.samp,  
  iter.update = 3,  
  iter.mcmc = 10000,  
  coef.nonzero,  
  var.coef.nonzero,
```

```

coef.zero,
var.coef.zero,
thin = 2,
burn.in = 2000,
tau.u.nZ = 1,
data
)

```

Arguments

formula	Formula that describe the fitted model
n.samp	Number of sample in each area
iter.update	Number of updates with default 3
iter.mcmc	Number of total iterations per chain with default 2000
coef.nonzero	Optional argument for mean on coefficient's prior distribution or β 's prior distribution which value is non-zero
var.coef.nonzero	Optional argument for the variances of the prior distribution of the model coefficients (β)
coef.zero	Optional argument for mean on coefficient's prior distribution or α 's prior distribution which value is non-zero
var.coef.zero	Optional argument for the variances of the prior distribution of the model coefficients (α)
thin	Thinning rate, must be a positive integer with default 1
burn.in	Number of iterations to discard at the beginning with default 1000
tau.u.nZ	Variance of random effect area for non-zero of variable interest (y) with default 1
data	The data frame

Value

This function returns a list of the following objects:

Est	A vector with the values of Small Area mean Estimates using Hierarchical bayesian method
refVar	Estimated random effect variances
coefficient	A dataframe with the estimated model coefficient
plot_alpha	Trace, Density, Autocorrelation Function Plot of MCMC samples
plot_beta	Trace, Density, Autocorrelation Function Plot of MCMC samples

Examples

```

#Compute Fitted Model
y ~ X1 +X2

# For data without any nonsampled area
# Load Dataset
data(dataZIB)
saeHB.ZIB <- ziBinomial(formula = y~X1+X2, "s", iter.update=3, iter.mcmc = 1000,
                        burn.in = 200,data = dataZIB)
#the setting of iter.update, iter.mcmc, and burn.in in this example
#is considered to make the example execution time be faster.
#Result
saeHB.ZIB$Est                                #Small Area mean Estimates
saeHB.ZIB$Est$SD                             #Standard deviation of Small Area Mean Estimates
saeHB.ZIB$refVar                             #refVar
saeHB.ZIB$coefficient                        #coefficient
#Load Library 'coda' to execute the plot
#autocorr.plot(saeHB.ZIB$plot_alpha[[3]]) is used to #ACF Plot for alpha
#autocorr.plot(saeHB.ZIB$plot_beta[[3]]) is used to #ACF Plot for beta
#plot(saeHB.ZIB$plot_alpha[[3]]) is used to #Dencity and trace plot for alpha
#plot(saeHB.ZIB$plot_beta[[3]]) is used to #Dencity and trace plot for beta

```

Index

* **datasets**

dataZIB, [2](#)

dataZIBns, [3](#)

dataZIB, [2](#)

dataZIBns, [3](#)

ziBinomial, [3](#)