Package ‘saeHB’

October 14, 2022

Type Package

Title Small Area Estimation using Hierarchical Bayesian Method

Version 0.2.1

Author Azka Ubaidillah [aut], Ika Yuni Wulansari [aut], Zaza Yuda Perwira [aut, cre]

Maintainer Zaza Yuda Perwira <221710086@stis.ac.id>

Description Provides several functions for area level of small area estimation using hierarchical Bayesian (HB) method with several univariate distributions for variable of interest. The dataset that used in every function is generated accordingly in the Example. 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. For the reference, see Rao and Molina (2015) <doi:10.1002/9781118735855>.

License GPL-3

Encoding UTF-8

RoxygenNote 7.1.1

URL https://github.com/zazaperwira/saeHB

BugReports https://github.com/zazaperwira/saeHB/issues

Suggests knitr, rmarkdown

VignetteBuilder knitr

Imports stringr, coda, rjags, stats, nimble, CARBayesdata, MASS, grDevices, graphics

SystemRequirements JAGS (http://mcmc-jags.sourceforge.net)

Depends R (>= 2.10)

NeedsCompilation no

Repository CRAN

Date/Publication 2022-04-08 15:10:02 UTC

R topics documented:

Beta *Small Area Estimation using Hierarchical Bayesian under Beta Distribution*

Description

This function is implemented to variable of interest (y) that assumed to be a Beta Distribution. The range of data must be $0 < y < 1$. The data proportion is supposed to be implemented with this function.

Usage

```
Beta(
  formula,
  iter.update = 3,
  iter.mcmc = 10000,
  coef,
  var.coef,
  thin = 3,
  burn.in = 2000,
  tau.u = 1,
  data
)
```

Arguments

`formula`	Formula that describe the fitted model
`iter.update`	Number of updates with default `3`
`iter.mcmc`	Number of total iterations per chain with default `10000`
`coef`	a vector contains prior initial value of Coefficient of Regression Model for fixed effect with default vector of `0` with the length of the number of regression coefficients
`var.coef`	a vector contains prior initial value of variance of Coefficient of Regression Model with default vector of `1` with the length of the number of regression coefficients
`thin`	Thinning rate, must be a positive integer with default `2`
`burn.in`	Number of iterations to discard at the beginning with default `2000`
`tau.u`	Prior initial value of inverse of Variance of area random effect 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`	Trace, Dencity, Autocorrelation Function Plot of MCMC samples

Examples

```
#Data Generation
set.seed(123)
m=30
x1=runif(m,0,1)
x2=runif(m,0,1)
x3=runif(m,0,1)
x4=runif(m,0,1)
b0=b1=b2=b3=b4=0.5
u=rnorm(m,0,1)
pi=rgamma(1,1,0.5)
Mu <- exp(b0+b1*x1+b2*x2+b3*x3+b4*x4+u)/(1+exp(b0+b1*x1+b2*x2+b3*x3+b4*x4+u))
A=Mu*pi
B=(1-Mu) * pi
y=rbeta(m,A,B)
y <- ifelse(y<1,y,0.99999999)
y <- ifelse(y>0,y,0.00000001)
MU=A/(A+B)
vardir=A*B/((A+B)^2*(A+B+1))
dataBeta = as.data.frame(cbind(y,x1,x2,x3,x4,vardir))
```

```
dataBetaNs=dataBeta
dataBetaNs$y[c(3,14,22,29,30)] <- NA
dataBetaNs$vardir[c(3,14,22,29,30)] <- NA


##Compute Fitted Model
##y ~ x1 +x2


## For data without any nonsampled area

#vc = c(1,1,1)
#c = c(0,0,0)
#formula = y~x1+x2
#dat = dataBeta[1:10,]


##Using parameter coef and var.coef
#saeHBbeta<-Beta(formula,var.coef=vc,iter.update=10,coef=c,data=dat)

#saeHBbeta$Est                                  #Small Area mean Estimates
#saeHBbeta$refVar                               #Random effect variance
#saeHBbeta$coefficient                          #coefficient
#Load Library 'coda' to execute the plot
#autocorr.plot(saeHBbeta$plot[[3]]) is used to generate ACF Plot
#plot(saeHBbeta$plot[[3]]) is used to generate Density and trace plot

##Do not use parameter coef and var.coef
#saeHBbeta <- Beta(formula,data=dataBeta)
```

Binomial — *Small Area Estimation using Hierarchical Bayesian under Binomial Distribution*

Description

This function is implemented to variable of interest (y) that assumed to be a Binomial Distribution using Logit normal model. The data is an accumulation from the Bernoulli process which there are exactly two mutually exclusive outcomes of a case.

Usage

```
Binomial(
  formula,
  n.samp,
  iter.update = 3,
  iter.mcmc = 10000,
  coef,
  var.coef,
  thin = 2,
  burn.in = 2000,
  tau.u = 1,
  data
)
```

Arguments

`formula`	Formula that describe the fitted model
`n.samp`	number of sample
`iter.update`	Number of updates with default `3`
`iter.mcmc`	Number of total iterations per chain with default `10000`
`coef`	a vector contains prior initial value of Coefficient of Regression Model for fixed effect with default vector of `0` with the length of the number of regression coefficients
`var.coef`	a vector contains prior initial value of variance of Coefficient of Regression Model with default vector of `1` with the length of the number of regression coefficients
`thin`	Thinning rate, must be a positive integer with default `2`
`burn.in`	Number of iterations to discard at the beginning with default `2000`
`tau.u`	Prior initial value of inverse of Variance of area random effect 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, if there is non sampled area, then the result of estimation of mu in nonsampled areas are the probabilites
`refVar`	Estimated random effect variances
`coefficient`	A dataframe with the estimated model coefficient
`plot`	Trace, Dencity, Autocorrelation Function Plot of MCMC samples

Author(s)

Azka Ubaidillah [aut], Ika Yuni Wulansari [aut], Zaza Yuda Perwira [aut, cre], Jayanti Wulansari [aut, cre], Fauzan Rais Arfizain [aut,cre]

Examples

```
#Data Generation
set.seed(123)
m=30
x1=runif(m,0,1)
```

```
x2=runif(m,0,1)
b0=b1=b2=0.5
u=rnorm(m,0,1)
n.samp1=round(runif(m,10,30))
mu= exp(b0 + b1*x1+b2*x2+u)/(1+exp(b0 + b1*x1+b2*x2+u))
y=rbinom(m,n.samp1,mu)
vardir=n.samp1*mu*(1-mu)
dataBinomial=as.data.frame(cbind(y,x1,x2,n.samp=n.samp1,vardir))
dataBinomialNs = dataBinomial
dataBinomialNs$y[c(3,14,22,29,30)] <- NA
dataBinomialNs$vardir[c(3,14,22,29,30)] <- NA
dataBinomialNs$n.samp[c(3,14,22,29,30)] <- NA


##Compute Fitted Model
##y~x1+x2


## For data without any nonsampled area
#formula = y~x1+x2
#n.s = "n.samp"
#vc = c(1,1,1)
#c = c(0,0,0)
#dat = dataBinomial


## Using parameter coef and var.coef
#saeHBBinomial<-Binomial(formula,n.samp=n.s,iter.update=10,coef=c,var.coef=vc,data =dat)

#saeHBBinomial$Est                                   #Small Area mean Estimates
#saeHBBinomial$refVar                                #Random effect variance
#saeHBBinomial$coefficient                           #coefficient
#Load Library 'coda' to execute the plot
#autocorr.plot(saeHBBinomial$plot[[3]]) is used to generate ACF Plot
#plot(saeHBBinomial$plot[[3]]) is used to generate Density and trace plot

## Do not using parameter coef and var.coef
#saeHBBinomial <- Binomial(formula,n.samp ="n.samp",data=dataBinomial)



## For data with nonsampled area use dataBinomialNs
```

Exponential *Small Area Estimation using Hierarchical Bayesian under Exponential Distribution*

Description

This function is implemented to variable of interest (y) that assumed to be a Exponential Distribution. The range of data is $y > 0)$

Usage

```
Exponential(
  formula,
  iter.update = 3,
  iter.mcmc = 10000,
  coef,
  var.coef,
  thin = 2,
  burn.in = 2000,
  tau.u = 1,
  data
)
```

Arguments

`formula`	Formula that describe the fitted model
`iter.update`	Number of updates with default `3`
`iter.mcmc`	Number of total iterations per chain with default `10000`
`coef`	a vector contains prior initial value of Coefficient of Regression Model for fixed effect with default vector of `0` with the length of the number of regression coefficients
`var.coef`	a vector contains prior initial value of variance of Coefficient of Regression Model with default vector of `1` with the length of the number of regression coefficients
`thin`	Thinning rate, must be a positive integer with default `2`
`burn.in`	Number of iterations to discard at the beginning with default `2000`
`tau.u`	Prior initial value of inverse of Variance of area random effect 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`	Trace, Dencity, Autocorrelation Function Plot of MCMC samples

Examples

```
#Data Generation
set.seed(123)
m=30
x1=runif(m,0,1)
x2=runif(m,0,1)
b0=b1=b2=0.5
u=rnorm(m,0,1)
lambda= exp(b0 + b1*x1 + b2*x2+u)
mu=1/lambda
y=rexp(m,lambda)
vardir=1/lambda^2
hist(y)
dataExp=as.data.frame(cbind(y,x1,x2,vardir))
dataExpNs <- dataExp
dataExpNs$y[c(3,14,22,29,30)] <- NA
dataExpNs$vardir[c(3,14,22,29,30)] <- NA


##Compute Fitted Model
##y ~ x1 +x2


## For data without any nonsampled area

#formula = y ~ x1+x2
#v = c(1,1,1)
#c = c(0,0,0)


## Using parameter coef and var.coef
#saeHBExponential <- Exponential(formula,coef=c,var.coef=v,iter.update=10,data=dataExp)

#saeHBExponential$Est                             #Small Area mean Estimates
#saeHBExponential$refVar                          #Random effect variance
#saeHBExponential$coefficient                     #coefficient
#Load Library 'coda' to execute the plot
#autocorr.plot(saeHBExponential$plot[[3]]) is used to generate ACF Plot
#plot(saeHBExponential$plot[[3]]) is used to generate Density and trace plot

## Do not using parameter coef and var.coef
#saeHBExponential <- Exponential(formula,data=dataExp[1:10,])



## For data with nonsampled area use dataExpNs
```

ExponentialDouble	*Small Area Estimation using Hierarchical Bayesian under Double Exponential Distribution*

Description

This function is implemented to variable of interest (y) that assumed to be a Double Exponential Distribution or Laplace Distribution. The range of data is $(-\infty < y < \infty)$

Usage

```
ExponentialDouble(
  formula,
  iter.update = 3,
  iter.mcmc = 10000,
  coef,
  var.coef,
  thin = 2,
  burn.in = 2000,
  tau.u = 1,
  data
)
```

Arguments

formula	Formula that describe the fitted model
iter.update	Number of updates with default 3
iter.mcmc	Number of total iterations per chain with default 10000
coef	a vector contains prior initial value of Coefficient of Regression Model for fixed effect with default vector of 0 with the length of the number of regression coefficients
var.coef	a vector contains prior initial value of variance of Coefficient of Regression Model with default vector of 1 with the length of the number of regression coefficients
thin	Thinning rate, must be a positive integer with default 2
burn.in	Number of iterations to discard at the beginning with default 2000
tau.u	Prior initial value of inverse of Variance of area random effect 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	Trace, Dencity, Autocorrelation Function Plot of MCMC samples

Examples

```
##Data Generation
set.seed(123)
library(nimble)
m=30
x1=runif(m,10,20)
x2=runif(m,1,10)
b0=b1=b2=0.5
u=rnorm(m,0,1)
tau=rgamma(m,1,1)
sd=1/sqrt(tau)
mu=b0 + b1*x1+b2*x2+u
y=rdexp(m,mu,sd)
vardir=sqrt(2)*sd^2
dataExpDouble=as.data.frame(cbind(y,x1,x2,vardir))
dataExpDoubleNs=dataExpDouble
dataExpDoubleNs$y[c(3,14,22,29,30)] <- NA
dataExpDoubleNs$vardir[c(3,14,22,29,30)] <- NA


##Compute Fitted Model
##y ~ x1 +x2


## For data without any nonsampled area
#formula = y ~ x1+x2
#vc = c(1,1,1)
#c = c(0,0,0)
#dat = dataExpDouble[1:10,]


## Using parameter coef and var.coef

#saeHBExpDouble<-ExponentialDouble(formula,coef=c,var.coef=vc,iter.update=10,data=dat)

#saeHBExpDouble$Est                                 #Small Area mean Estimates
#saeHBExpDouble$refVar                              #Random effect variance
#saeHBExpDouble$coefficient                         #coefficient
#Load Library 'coda' to execute the plot
#autocorr.plot(saeHBExpDouble$plot[[3]]) is used to generate ACF Plot
#plot(saeHBExpDouble$plot[[3]]) is used to generate Density and trace plot

## Do not using parameter coef and var.coef
#saeHBExpDouble <- ExponentialDouble(formula,data=dataExpDouble)
```

```
## For data with nonsampled area use dataExpDoubleNs
```

`Gamma` *Small Area Estimation using Hierarchical Bayesian under Gamma Distribution*

Description

This function is implemented to variable of interest (y) that assumed to be a Gamma Distribution. The range of data is $(y > 0)$

Usage

```
Gamma(
  formula,
  iter.update = 3,
  iter.mcmc = 10000,
  coef,
  var.coef,
  thin = 2,
  burn.in = 2000,
  tau.u = 1,
  data
)
```

Arguments

`formula`	Formula that describe the fitted model
`iter.update`	Number of updates with default `3`
`iter.mcmc`	Number of total iterations per chain with default `10000`
`coef`	a vector contains prior initial value of Coefficient of Regression Model for fixed effect with default vector of `0` with the length of the number of regression coefficients
`var.coef`	a vector contains prior initial value of variance of Coefficient of Regression Model with default vector of `1` with the length of the number of regression coefficients
`thin`	Thinning rate, must be a positive integer with default `2`
`burn.in`	Number of iterations to discard at the beginning with default `2000`
`tau.u`	Prior initial value of inverse of Variance of area random effect 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`	Trace, Dencity, Autocorrelation Function Plot of MCMC samples

Examples

```
##Data Generation
set.seed(123)
m=30
x1=runif(m,0,1)
x2=runif(m,0,1)
b0=b1=b2=0.5
u=rnorm(m,0,1)
phi=rgamma(m,0.5,0.5)
vardir=1/phi
mu= exp(b0 + b1*x1+b2*x2+u)
A=mu^2*phi
B=mu*phi
y=rgamma(m,A,B)

dataGamma=as.data.frame(cbind(y,x1,x2,vardir))
dataGammaNs <- dataGamma
dataGammaNs$y[c(3,14,22,29,30)] <- NA
dataGammaNs$vardir[c(3,14,22,29,30)] <- NA


##Compute Fitted Model
##y ~ x1 +x2


## For data without any nonsampled area
#formula = y ~ x1 +x2
#v = c(1,1,1)
#c = c(0,0,0)


## Using parameter coef and var.coef
#saeHBGamma <- Gamma(formula,coef=c,var.coef=v,iter.update=10,data =dataGamma)

#saeHBGamma$Est                                #Small Area mean Estimates
#saeHBGamma$refVar                             #Random effect variance
#saeHBGamma$coefficient                        #coefficient
#Load Library 'coda' to execute the plot
#autocorr.plot(saeHBGamma$plot[[3]]) is used to generate ACF Plot
#plot(saeHBGamma$plot[[3]]) is used to generate Density and trace plot
```

```
## Do not using parameter coef and var.coef
#saeHBGamma <- Gamma(formula, data =  dataGamma)#'


## For data with nonsampled area use dataGammaNs
```

`Logistic` *Small Area Estimation using Hierarchical Bayesian under Logistic Distribution*

Description

This function is implemented to variable of interest (y) that assumed to be a Logistic Distribution

Usage

```
Logistic(
  formula,
  iter.update = 3,
  iter.mcmc = 10000,
  coef,
  var.coef,
  thin = 2,
  burn.in = 2000,
  tau.u = 1,
  data
)
```

Arguments

`formula`	Formula that describe the fitted model
`iter.update`	Number of updates with default `3`
`iter.mcmc`	Number of total iterations per chain with default `10000`
`coef`	a vector contains prior initial value of Coefficient of Regression Model for fixed effect with default vector of `0` with the length of the number of regression coefficients
`var.coef`	a vector contains prior initial value of variance of Coefficient of Regression Model with default vector of `1` with the length of the number of regression coefficients
`thin`	Thinning rate, must be a positive integer with default `2`
`burn.in`	Number of iterations to discard at the beginning with default `2000`
`tau.u`	Prior initial value of inverse of Variance of area random effect 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`	Trace, Dencity, Autocorrelation Function Plot of MCMC samples

Examples

```
##Data Generation
set.seed(123)
m=30
x1=runif(m,0,1)
x2=runif(m,1,2)
x3=runif(m,2,3)
b0=b1=b2=b3=0.5
u=rnorm(m,0,1)
Mu=b0 + b1*x1+b2*x2+b3*x3+u
sig=sqrt(1/rgamma(m,1,1))
y=rlogis(m,Mu,sig)
vardir=1/3*pi*sig^2
dataLogistic=as.data.frame(cbind(y,x1,x2,x3,vardir))
dataLogisticNs=dataLogistic
dataLogisticNs$y[c(3,14,22,29,30)] <- NA
dataLogisticNs$vardir[c(3,14,22,29,30)] <- NA


##Compute Fitted Model
##y ~ x1 +x2 +x3


## For data without any nonsampled area
#formula = y ~ x1 +x2 +x3
#v = c(1,1,1,1)
#c = c(0,0,0,0)


## Using parameter coef and var.coef
#saeHBLogistic <- Logistic(formula,coef=c,var.coef=v,iter.update=10,data =dataLogistic)

#saeHBLogistic$Est                                   #Small Area mean Estimates
#saeHBLogistic$refVar                                #Random effect variance
#saeHBLogistic$coefficient                           #coefficient
#Load Library 'coda' to execute the plot
#autocorr.plot(saeHBLogistic$plot[[3]]) is used to generate ACF Plot
#plot(saeHBLogistic$plot[[3]]) is used to generate Density and trace plot

## Do not using parameter coef and var.coef
```

```
#saeHBLogistic <- Logistic(formula,data =dataLogistic)


## For data with nonsampled area use dataLogisticNs
```

`Lognormal` *Small Area Estimation using Hierarchical Bayesian under Lognormal Distribution*

Description

This function is implemented to variable of interest (y) that assumed to be a Lognormal Distribution. The range of data is $(y > 0$

Usage

```
Lognormal(
  formula,
  iter.update = 3,
  iter.mcmc = 10000,
  coef,
  var.coef,
  thin = 2,
  burn.in = 2000,
  tau.u = 1,
  data
)
```

Arguments

`formula`	Formula that describe the fitted model
`iter.update`	Number of updates with default `3`
`iter.mcmc`	Number of total iterations per chain with default `10000`
`coef`	a vector contains prior initial value of Coefficient of Regression Model for fixed effect with default vector of `0` with the length of the number of regression coefficients
`var.coef`	a vector contains prior initial value of variance of Coefficient of Regression Model with default vector of `1` with the length of the number of regression coefficients
`thin`	Thinning rate, must be a positive integer with default `2`
`burn.in`	Number of iterations to discard at the beginning with default `2000`
`tau.u`	Prior initial value of inverse of Variance of area random effect 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`	Trace, Dencity, Autocorrelation Function Plot of MCMC samples

Examples

```
##Data Generation
set.seed(123)
m=30
x1=runif(m,0,1)
x2=runif(m,1,2)
x3=runif(m,2,3)
b0=b1=b2=b3=0.5
u=rnorm(m,0,1)
mu=b0 + b1*x1+b2*x2+b3*x3+u
sig=1
y=rlnorm(m,mu,sig)
E=exp(mu+1/2*sig^2)
vardir=exp(2*mu+sig^2)*(exp(sig^2)-1)
dataLognormal=as.data.frame(cbind(y,x1,x2,x3,vardir))
dataLognormalNs=dataLognormal
dataLognormalNs$y[c(3,14,22,29,30)] <- NA
dataLognormalNs$vardir[c(3,14,22,29,30)] <- NA


##Compute Fitted Model
##y ~ x1 +x2 +x3


## For data without any nonsampled area

#formula = y ~ x1 +x2 +x3
#v = c(1,1,1,1)
#c= c(0,0,0,0)


## Using parameter coef and var.coef
#saeHBLognormal <- Lognormal(formula,coef=c,var.coef=v,iter.update=10,data=dataLognormal)

#saeHBLognormal$Est                                    #Small Area mean Estimates
#saeHBLognormal$refVar                                 #Random effect variance
#saeHBLognormal$coefficient                            #coefficient
#Load Library 'coda' to execute the plot
#autocorr.plot(saeHBLognormal$plot[[3]]) is used to generate ACF Plot
#plot(saeHBLognormal$plot[[3]]) is used to generate Density and trace plot
```

```
## Do not using parameter coef and var.coef
#saeHBLognormal <- Lognormal(formula,data=dataLognormal)


## For data with nonsampled area use dataLognormalNs
```

`NegativeBinomial` *Small Area Estimation using Hierarchical Bayesian under Negative Binomial Distribution*

Description

This function is implemented to variable of interest (y) that assumed to be a Negative Binomial Distribution. The data is a number of the Bernoulli process. The negative binomial is used to overcome an over dispersion from the discrete model.

Usage

```
NegativeBinomial(
  formula,
  iter.update = 3,
  iter.mcmc = 10000,
  coef,
  var.coef,
  thin = 2,
  burn.in = 2000,
  tau.u = 1,
  data
)
```

Arguments

`formula`	Formula that describe the fitted model
`iter.update`	Number of updates with default `3`
`iter.mcmc`	Number of total iterations per chain with default `10000`
`coef`	a vector contains prior initial value of Coefficient of Regression Model for fixed effect with default vector of `0` with the length of the number of regression coefficients
`var.coef`	a vector contains prior initial value of variance of Coefficient of Regression Model with default vector of `1` with the length of the number of regression coefficients
`thin`	Thinning rate, must be a positive integer with default `2`
`burn.in`	Number of iterations to discard at the beginning with default `2000`
`tau.u`	Prior initial value of inverse of Variance of area random effect 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`	Trace, Dencity, Autocorrelation Function Plot of MCMC samples

Examples

```
##Data Generation
set.seed(123)
library(MASS)
m=30
x=runif(m,0,1)
b0=b1=0.5
u=rnorm(m,0,1)
Mu=exp(b0+b1*x+u)
theta=1
y=rnegbin(m,Mu,theta)
vardir=Mu+Mu^2/theta
dataNegativeBinomial=as.data.frame(cbind(y,x,vardir))
dataNegativeBinomialNs=dataNegativeBinomial
dataNegativeBinomialNs$y[c(3,14,22,29,30)] <- NA
dataNegativeBinomialNs$vardir[c(3,14,22,29,30)] <- NA


## Compute Fitted Model
## y ~ x


## For data without any nonsampled area

#formula = y ~ x
#v= c(1,1)
#c= c(0,0)
#dat = dataNegativeBinomial

## Using parameter coef and var.coef
#saeHBNegbin <- NegativeBinomial(formula,coef=c,var.coef=v,iter.update=10,data =dat)

#saeHBNegbin$Est                                 #Small Area mean Estimates
#saeHBNegbin$refVar                              #Random effect variance
#saeHBNegbin$coefficient                         #coefficient
#Load Library 'coda' to execute the plot
#autocorr.plot(saeHBNegbin$plot[[3]]) is used to generate ACF Plot
#plot(saeHBNegbin$plot[[3]]) is used to generate Density and trace plot

## Do not using parameter coef and var.coef
```

```
#saeHBNegbin <- NegativeBinomial(formula,data =dat)



## For data with nonsampled area use dataNegativeBinomialNs
```

Normal — *Small Area Estimation using Hierarchical Bayesian under Normal Distribution*

Description

This function is implemented to variable of interest (y) that assumed to be a Normal Distribution. The range of data is $(-\infty < y < \infty)$

Usage

```
Normal(
  formula,
  vardir,
  iter.update = 3,
  iter.mcmc = 10000,
  coef,
  var.coef,
  thin = 2,
  burn.in = 2000,
  tau.u = 1,
  data
)
```

Arguments

`formula`	Formula that describe the fitted model
`vardir`	Sampling variances of direct estimations
`iter.update`	Number of updates with default `3`
`iter.mcmc`	Number of total iterations per chain with default `10000`
`coef`	a vector contains prior initial value of Coefficient of Regression Model for fixed effect with default vector of `0` with the length of the number of regression coefficients
`var.coef`	a vector contains prior initial value of variance of Coefficient of Regression Model with default vector of `1` with the length of the number of regression coefficients
`thin`	Thinning rate, must be a positive integer with default `2`
`burn.in`	Number of iterations to discard at the beginning with default `2000`
`tau.u`	Prior initial value of inverse of Variance of area random effect 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`	Trace, Dencity, Autocorrelation Function Plot of MCMC samples

Examples

```
#Data Generation
set.seed(123)
m=30
x1=runif(m,0,1)
x2=runif(m,1,5)
x3=runif(m,10,15)
x4=runif(m,10,20)
b0=b1=b2=b3=b4=0.5
u=rnorm(m,0,1)
vardir=1/rgamma(m,1,1)
Mu <- b0+b1*x1+b2*x2+b3*x3+b4*x4+u
y=rnorm(m,Mu,sqrt(vardir))
dataNormal=as.data.frame(cbind(y,x1,x2,x3,x4,vardir))
dataNormalNs=dataNormal
dataNormalNs$y[c(3,10,15,29,30)] <- NA
dataNormalNs$vardir[c(3,10,15,29,30)] <- NA


##Compute Fitted Model
##y ~ x1 +x2 +x3 +x4


## For data without any nonsampled area
#formula=y~x1+x2+x3+x4
#var= "vardir"
#v = c(1,1,1,1,1)
#c= c(0,0,0,0,0)


## Using parameter coef and var.coef
#saeHBnormal<-Normal(formula,vardir=var,coef=c,var.coef=v,data=dataNormal)

#saeHBnormal$Est                                 #Small Area mean Estimates
#saeHBnormal$refVar                              #Random effect variance
#saeHBnormal$coefficient                         #coefficient
#Load Library 'coda' to execute the plot
#autocorr.plot(saeHBnormal$plot[[3]]) is used to generate ACF Plot
#plot(saeHBnormal$plot[[3]]) is used to generate Density and trace plot
```

```
## Do not using parameter coef and var.coef
#saeHBnormal<-Normal(formula,vardir ="vardir",data=dataNormal)



## For data with nonsampled area use dataNormalNs
```

`Poisson` *Small Area Estimation using Hierarchical Bayesian under Poisson Distribution*

Description

This function is implemented to variable of interest (y) that assumed to be a Poisson Distribution. The data is a count data, $y = 1, 2, 3, ...$

Usage

```
Poisson(
  formula,
  iter.update = 3,
  iter.mcmc = 10000,
  coef,
  var.coef,
  thin = 2,
  burn.in = 2000,
  tau.u = 1,
  data
)
```

Arguments

`formula`	Formula that describe the fitted model
`iter.update`	Number of updates with default `3`
`iter.mcmc`	Number of total iterations per chain with default `10000`
`coef`	a vector contains prior initial value of Coefficient of Regression Model for fixed effect with default vector of `0` with the length of the number of regression coefficients
`var.coef`	a vector contains prior initial value of variance of Coefficient of Regression Model with default vector of `1` with the length of the number of regression coefficients
`thin`	Thinning rate, must be a positive integer with default `2`
`burn.in`	Number of iterations to discard at the beginning with default `2000`
`tau.u`	Prior initial value of inverse of Variance of area random effect 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`	Trace, Dencity, Autocorrelation Function Plot of MCMC samples

Author(s)

Azka Ubaidillah [aut], Ika Yuni Wulansari [aut], Zaza Yuda Perwira [aut, cre], Jayanti Wulansari [aut, cre], Fauzan Rais Arfizain [aut,cre]

Examples

```
##Load Dataset
library(CARBayesdata)
data(lipdata)
dataPoisson <- lipdata
dataPoissonNs <- lipdata
dataPoissonNs$observed[c(2,9,15,23,40)] <- NA


##Compute Fitted Model
##observed ~ pcaff


## For data without any nonsampled area

#formula = observed ~ pcaff
#v = c(1,1)
#c = c(0,0)


## Using parameter coef and var.coef
#saeHBPoisson <- Poisson(formula, coef=c,var.coef=v,iter.update=10,data=dataPoisson)

#saeHBPoisson$Est                                  #Small Area mean Estimates
#saeHBPoisson$refVar                               #Random effect variance
#saeHBPoisson$coefficient                          #coefficient
#Load Library 'coda' to execute the plot
#autocorr.plot(saeHBPoisson$plot[[3]]) is used to generate ACF Plot
#plot(saeHBPoisson$plot[[3]]) is used to generate Density and trace plot

## Do not using parameter coef and var.coef
#saeHBPoisson <- Poisson(formula,data=dataPoisson)
```

```
## For data with nonsampled area use dataPoissonNs
```

PoissonGamma *Small Area Estimation using Hierarchical Bayesian under Poisson Gamma Distribution*

Description

This function is implemented to variable of interest (y) that assumed to be a Poisson Distribution which it is parameter (λ) is assumed to be a Gamma distribution. The data is a count data, $y = 1, 2, 3, ...$

Usage

```
PoissonGamma(
  formula,
  iter.update = 3,
  iter.mcmc = 10000,
  coef,
  var.coef,
  thin = 2,
  burn.in = 2000,
  tau.u = 1,
  data
)
```

Arguments

`formula`	Formula that describe the fitted model
`iter.update`	Number of updates with default `3`
`iter.mcmc`	Number of total iterations per chain with default `10000`
`coef`	a vector contains prior initial value of Coefficient of Regression Model for fixed effect with default vector of `0` with the length of the number of regression coefficients
`var.coef`	a vector contains prior initial value of variance of Coefficient of Regression Model with default vector of `1` with the length of the number of regression coefficients
`thin`	Thinning rate, must be a positive integer with default `2`
`burn.in`	Number of iterations to discard at the beginning with default `2000`
`tau.u`	Prior initial value of inverse of Variance of area random effect 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`	Trace, Dencity, Autocorrelation Function Plot of MCMC samples

Examples

```
##Load Dataset
library(CARBayesdata)
data(lipdata)
dataPoissonGamma <- lipdata
dataPoissonGammaNs <- lipdata
dataPoissonGammaNs$observed[c(2,9,15,23,40)] <- NA


##Compute Fitted Model
##observed ~ pcaff


## For data without any nonsampled area

#formula = observed ~ pcaff
#v= c(1,1)
#c = c(0,0)
#dat = dataPoissonGamma


## Using parameter coef and var.coef
#saeHBPoissonGamma <- PoissonGamma(formula,coef=c,var.coef=v,iter.update=10,data=dat)

#saeHBPoissonGamma$Est                              #Small Area mean Estimates
#saeHBPoissonGamma$refVar                           #Random effect variance
#saeHBPoissonGamma$coefficient                      #coefficient
#Load Library 'coda' to execute the plot
#autocorr.plot(saeHBPoissonGamma$plot[[3]]) is used to generate ACF Plot
#plot(saeHBPoissonGamma$plot[[3]]) is used to generate Density and trace plot

## Do not using parameter coef and var.coef
#saeHBPoissonGamma <- PoissonGamma(formula,data=dataPoissonGamma)



## For data with nonsampled area use dataPoissonGammaNs
```

`saeHB` *saeHB : Small Area Estimation using Hierarchical Bayesian Method*

Description

Provides several functions for area level of small area estimation using hierarchical Bayesian (HB) method with several univariate distributions for variable of interest. The dataset that used in every function is generated accordingly in the Example. 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. For the reference, see Rao and Molina (2015) <doi:10.1002/9781118735855>.

Author(s)

Azka Ubaidillah <`azka@stis.ac.id`>, Ika Yuni Wulansari <`ikayuni@stis.ac.id`>, Zaza Yuda Perwira <`221710086@stis.ac.id`>

Maintainer: Zaza Yuda Perwira <`221710086@stis.ac.id`>

Functions

`Beta` Produces HB estimators, standard error, random effect variance, coefficient and plot under beta distribution

`Binomial` Produces HB estimators, standard error, random effect variance, coefficient and plot under binomial distribution

`Exponential` Produces HB estimators, standard error, random effect variance, coefficient and plot under exponential distribution

`ExponentialDouble` Produces HB estimators, standard error, random effect variance, coefficient and plot double exponential distribution

`Gamma` Produces HB estimators, standard error, random effect variance, coefficient and plot under Gamma distribution

`Logistic` Produces HB estimators, standard error, random effect variance, coefficient and plot under logistic distribution

`Lognormal` Produces HB estimators, standard error, random effect variance, coefficient and plot under lognormal distribution

`NegativeBinomial` Produces HB estimators, standard error, random effect variance, coefficient and plot under negative binomial distribution

`Normal` Produces HB estimators, standard error, random effect variance, coefficient and plot under normal distribution

`Poisson` Produces HB estimators, standard error, random effect variance, coefficient and plot under poisson distribution

`PoissonGamma` Produces HB estimators, standard error, random effect variance, coefficient and plot under poisson gamma distribution

`Student_t` Produces HB estimators, standard error, random effect variance, coefficient and plot under student-t distribution

Student_tnc	Produces HB estimators, standard error, random effect variance, coefficient and plot under student-t (not central) distribution
Weibull	Produces HB estimators, standard error, random effect variance, coefficient and plot under weibull distribution

Reference

- Rao, J.N.K & Molina. (2015). Small Area Estimation 2nd Edition. New York: John Wiley and Sons, Inc. <doi:10.1002/9781118735855>.

Student_t *Small Area Estimation using Hierarchical Bayesian under Student-t Distribution*

Description

This function is implemented to variable of interest (y) that assumed to be a Student-t Distribution. The range of data is $(-\infty < y < \infty)$

Usage

```
Student_t(
  formula,
  iter.update = 3,
  iter.mcmc = 10000,
  coef,
  var.coef,
  thin = 2,
  burn.in = 2000,
  tau.u = 1,
  data
)
```

Arguments

formula	Formula that describe the fitted model
iter.update	Number of updates with default 3
iter.mcmc	Number of total iterations per chain with default 10000
coef	a vector contains prior initial value of Coefficient of Regression Model for fixed effect with default vector of 0 with the length of the number of regression coefficients
var.coef	a vector contains prior initial value of variance of Coefficient of Regression Model with default vector of 1 with the length of the number of regression coefficients
thin	Thinning rate, must be a positive integer with default 2

`burn.in`	Number of iterations to discard at the beginning with default `2000`
`tau.u`	Prior initial value of inverse of Variance of area random effect 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`	Trace, Dencity, Autocorrelation Function Plot of MCMC samples

Examples

```
##Data Generation
set.seed(123)
m=30
x1=runif(m,10,20)
x2=runif(m,30,50)
b0=b1=b2=0.5
u=rnorm(m,0,1)
MU=b0+b1*x1+b2*x2+u
k=rgamma(1,10,1)
y=rt(m,k,MU)
vardir=k/(k-1)
vardir=sd(y)^2
datat=as.data.frame(cbind(y,x1,x2,vardir))
datatNs=datat
datatNs$y[c(3,14,22,29,30)] <- NA
datatNs$vardir[c(3,14,22,29,30)] <- NA


##Compute Fitted Model
##y ~ x1 +x2


## For data without any nonsampled area

#formula =  y ~ x1+x2
#var.coef = c(1,1,1)
#coef = c(0,0,0)


## Using parameter coef and var.coef
#saeHBt <- Student_t(formula,coef=coef,var.coef=var.coef,iter.update=10,data = datat)

#saeHBt$Est                                #Small Area mean Estimates
#saeHBt$refVar                             #Random effect variance
```

```
#saeHBt$coefficient                        #coefficient
#Load Library 'coda' to execute the plot
#autocorr.plot(saeHBt$plot[[3]]) is used to generate ACF Plot
#plot(saeHBt$plot[[3]]) is used to generate Density and trace plot

## Do not using parameter coef and var.coef
#saeHBt <- Student_t(formula,data = datat)



## For data with nonsampled area use datatNs
```

`Student_tnc` *Small Area Estimation using Hierarchical Bayesian under Student-t (non central) Distribution*

Description

This function is implemented to variable of interest (y) that assumed to be a Student-t (non central) Distribution. The range of data is $(-\infty < y < \infty)$

Usage

```
Student_tnc(
  formula,
  iter.update = 3,
  iter.mcmc = 10000,
  coef,
  var.coef,
  thin = 2,
  burn.in = 2000,
  tau.u = 1,
  data
)
```

Arguments

`formula`	Formula that describe the fitted model
`iter.update`	Number of updates with default `3`
`iter.mcmc`	Number of total iterations per chain with default `10000`
`coef`	a vector contains prior initial value of Coefficient of Regression Model for fixed effect with default vector of `0` with the length of the number of regression coefficients
`var.coef`	a vector contains prior initial value of variance of Coefficient of Regression Model with default vector of `1` with the length of the number of regression coefficients

`thin`	Thinning rate, must be a positive integer with default `2`
`burn.in`	Number of iterations to discard at the beginning with default `2000`
`tau.u`	Prior initial value of inverse of Variance of area random effect 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`	Trace, Dencity, Autocorrelation Function Plot of MCMC samples

Examples

```
##Data Generation
set.seed(123)
m=30
x1=runif(m,1,100)
x2=runif(m,10,15)
b0=b1=b2=0.5
u=rnorm(m,0,1)
MU=b0+b1*x1+b2*x2+u
k=rgamma(1,10,1)
y=rt(m,k,MU)
vardir=k/(k-1)
vardir=sd(y)^2
datatnc=as.data.frame(cbind(y,x1,x2,vardir))
datatncNs=datatnc
datatncNs$y[c(3,14,22,29,30)] <- NA
datatncNs$vardir[c(3,14,22,29,30)] <- NA


##Compute Fitted Model
##y ~ x1 +x2


## For data without any nonsampled area

#formula =  y ~ x1+x2
#v = c(1,1,1)
#c = c(0,0,0)
#dat =   datatnc


## Using parameter coef and var.coef
#saeHBtnc <- Student_tnc(formula,coef=c, var.coef=v,iter.update=10, data = dat)
```

```
#saeHBtnc$Est                               #Small Area mean Estimates
#saeHBtnc$refVar                            #Random effect variance
#saeHBtnc$coefficient                       #coefficient
#Load Library 'coda' to execute the plot
#autocorr.plot(saeHBtnc$plot[[3]]) is used to generate ACF Plot
#plot(saeHBtnc$plot[[3]]) is used to generate Density and trace plot

## Do not using parameter coef and var.coef
#saeHBtnc <- Student_tnc(formula,data =  datatnc)



## For data with nonsampled area use datatncNs
```

Weibull — *Small Area Estimation using Hierarchical Bayesian under Weibull Distribution*

Description

This function is implemented to variable of interest (y) that assumed to be a Weibull Distribution. The range of data is $(y > 0$

Usage

```
Weibull(
  formula,
  iter.update = 3,
  iter.mcmc = 10000,
  coef,
  var.coef,
  thin = 2,
  burn.in = 2000,
  tau.u = 1,
  data
)
```

Arguments

`formula`	Formula that describe the fitted model
`iter.update`	Number of updates with default `3`
`iter.mcmc`	Number of total iterations per chain with default `10000`
`coef`	a vector contains prior initial value of Coefficient of Regression Model for fixed effect with default vector of `0` with the length of the number of regression coefficients

`var.coef`	a vector contains prior initial value of variance of Coefficient of Regression Model with default vector of `1` with the length of the number of regression coefficients
`thin`	Thinning rate, must be a positive integer with default `2`
`burn.in`	Number of iterations to discard at the beginning with default `2000`
`tau.u`	Prior initial value of inverse of Variance of area random effect 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`	Trace, Dencity, Autocorrelation Function Plot of MCMC samples

Examples

```
##Data Generation
set.seed(123)
m=30
x=runif(m,0,1)
b0=b1=0.5
u=rnorm(m,0,1)
Mu=exp(b0+b1*x+u)
k=rgamma(m,2,1)
lambda=Mu/gamma(1+1/k)
y=rweibull(m,k,lambda)
MU=lambda*gamma(1+1/k)
vardir=lambda^2*(gamma(1+2/k)-(gamma(1+1/k))^2)
dataWeibull=as.data.frame(cbind(y,x,vardir))
dataWeibullNs=dataWeibull
dataWeibullNs$y[c(3,14,22,29,30)] <- NA
dataWeibullNs$vardir[c(3,14,22,29,30)] <- NA


##Compute Fitted Model
##y ~ x


## For data without any nonsampled area

#formula = y ~ x
#var.coef = c(1,1)
#coef = c(0,0)
```

```
## Using parameter coef and var.coef
#saeHBWeibull <- Weibull(formula,coef=coef,var.coef=var.coef,iter.update=10,data=dataWeibull)

#saeHBWeibull$Est                                #Small Area mean Estimates
#saeHBWeibull$refVar                             #Random effect variance
#saeHBWeibull$coefficient                        #coefficient
#Load Library 'coda' to execute the plot
#autocorr.plot(saeHBWeibull$plot[[3]]) is used to generate ACF Plot
#plot(saeHBWeibull$plot[[3]]) is used to generate Density and trace plot

## Do not using parameter coef and var.coef
#saeHBWeibull <- Weibull(formula, data=dataWeibull)



## For data with nonsampled area use dataWeibullNs
```

Index