

# Package ‘zoib’

July 21, 2025

**Type** Package

**Title** Bayesian Inference for Beta Regression and Zero-or-One Inflated  
Beta Regression

**Version** 1.6

**Date** 2023-05-21

**Author** Fang Liu <fang.liu.131@nd.edu> with contributions from Yunchuan Kong

**Maintainer** Fang Liu <fang.liu.131@nd.edu>

**Depends** R (>= 3.3.0), rjags, coda, Formula, abind

**Description** Fits beta regression and zero-or-one inflated beta regression and obtains Bayesian Inference of the model via the Markov Chain Monte Carlo approach implemented in JAGS.

**URL** <https://www.r-project.org>, <https://www3.nd.edu/~fliu2/#tools>

**License** GPL (>= 3)

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2023-05-21 19:00:02 UTC

## Contents

zoib-package . . . . .	2
AlcoholUse . . . . .	3
BiRepeated . . . . .	4
check.psr . . . . .	5
GasolineYield . . . . .	6
paraplot . . . . .	8
pred.zoib . . . . .	9
zoib . . . . .	10

<b>Index</b>	<b>16</b>
--------------	-----------

---

zoib-package*Bayesian Inference for Beta Regression and Zero-or-One-Inflated Beta Regression Models*

---

## Description

beta regression and zero/one inflated beta regression on data with [0,1]-bounded responses. Bayesian inferences of the model are obtained via Markov Chain Monte Carlo sampling.

Package: zoib  
Type: Package  
Version: 1.6  
Date: 2023-05-21  
License: GPL (>= 3)

## Details

zoib accommodates boundary inflations at 0 or 1 of the response variables. It models clustered and correlated responses by introducing random components into the linear predictors of the link functions. The inferences from the models are based in the Bayesian framework via the Markov Chain Monte Carlo (MCMC) approaches.

The main function `zoib()` generates posterior samples on the regression coefficients, the design matrices in the link functions of the zoib regressions models, and predicted responses. It also produces a MCMC model object(JAGS) and posterior samples of the model parameters in the "mcmc" format. The DIC of the implemented Bayesian model can be calculated using `dic.samples(JAGS.object)` available in package `rjags` for model comparison purposes. Convergence of MCMC chains can be checked using functions `traceplot()`, `autocorr.plot()` and `gelman.diag()` available in package `coda` on the posterior draws from the MCMC algorithms. `zoib` also contains a function `check.psr()` that checks whether the multivariate psrf value can be calculated for multi-dimensional variables, provides box plots and summary statistics on multiple univariate psrf values. Posterior summary of the parameters can be obtained using `summary()`.

## Author(s)

Fang Liu (<[fang.liu.131@nd.edu](mailto:fang.liu.131@nd.edu)>), with contributions from Yunchuan Kong

Maintainer: Fang Liu (<[fang.liu.131@nd.edu](mailto:fang.liu.131@nd.edu)>)

## References

Liu, F. and Li, Q. (2014). A Bayesian Model for Joint Analysis of Multivariate Repeated Measures and Time to Event Data in Crossover Trials, *Statistical Methods in Medical Research*, DOI: 10.1177/0962280213519594

Liu, F. and Kong, Y. (2015). ZOIB: an R Package for Bayesian Inferences in Beta and Zero One Inflated Beta Regression Models, *The R Journal*, 7(2):34-51

Liu, F. and Eugenio, E.(2016). A review and comparison of Bayesian and likelihood-based inferences in beta regression and zero-or-one-inflated beta regression, *Statistical Methods in Medical Research*, DOI: 10.1177/0962280216650699

### See Also

betareg, rjags, coda

---

AlcoholUse

*California County-level Teenager Monthly Alcohol Use data*

---

### Description

AlcoholUse contains the county-level monthly alcohol use data from students in California in years 2008 to 2010. The data can be downloaded at <http://www.kidsdata.org>. The data has information on the percentage of public school students in grades 7, 9, and 11 in five buckets of days (0, 1-2, 3-9, 10-19, 20-30) in which they drank alcohol in the past 30 days. Since the percentages across the five days categories add up to 1 given a combination of County, Grade and Gender. Data points associated with 0 are completely redundant and are not in AlcoholUse. zoib is applied to examining whether the proportions of alcohol use in the past month are different across gender, grade, days of drinking.

### Usage

```
data(AlcoholUse)
```

### Format

A data frame with 1340 observations on the following 6 variables.

County a factor with 56 levels/clusters/counties.

Grade a numeric/factor vector: 7, 9, 11 grades.

Days a factor with levels 0 [1,2] [3,9] [10,19] [20,30].

MedDays a numeric vector: the med point of each of the 4 day buckets.

Gender a factor with levels Female Male.

Percentage a numeric vector; Proportions of public school students in 4 buckets of days in which they drank alcohol in the past 30 days per Gender, Grader, and County.

### Source

<http://www.kidsdata.org>

## Examples

```
## Not run:
##### eg3: modelling with clustered beta variables with inflation at 0
library(zoib)
data("AlcoholUse", package = "zoib")
eg3 <- zoib(Percentage ~ Grade*Gender+MedDays|1|Grade*Gender+MedDays|1,
            data = AlcoholUse, random = 1, EUID= AlcoholUse$County,
            zero.inflation = TRUE, one.inflation = FALSE, joint = FALSE,
            n.iter=5000, n.thin=20, n.burn=1000)

sample1 <- eg3$coeff
summary(sample1)

# check convergence on the regression coefficients
traceplot(sample1);
autocorr.plot(sample1);
check.psrfr(sample1)

# plot posterior mean of y vs. observed y to check on goodness of fit.
ypred = rbind(eg3$ypred[[1]],eg3$ypred[[2]])
post.mean= apply(ypred,2,mean);
par(mfrow=c(1,1),mar=c(4,4,0.5,0.5))
plot(AlcoholUse$Percentage, post.mean, xlim=c(0,0.4),ylim=c(0,0.4),
     col='blue', xlab='Observed y', ylab='Predicted y', main='')
abline(0,1,col='red')

## End(Not run)
```

---

BiRepeated

---

*Data from a correlated bivariate beta distribution with repeated measures*


---

## Description

BiRepeated is a simulated data set with bi-Beta variables  $y_1$  and  $y_2$  on 200 independent cases. Both  $y_1$  and  $y_2$  are repeatedly measured at a given set of covariate values  $x = (0.1, 0.2, 0.3, 0.4, 0.5, 0.6)$ . Joint zoib modeling is applied to examine the effect of  $x$  on  $y_1$  and  $y_2$  simultaneously.

## Usage

```
data(BiRepeated)
```

## Format

A data frame with 200 independent cases, from which 6 measurements are taken on 2 response variables.

**id** id of the 200 cases.

**y1** value of one beta variable (6 measurements per case) ranged from 0 to 1.

**y2** value of the other beta variables (6 measurements per case) ranged from 0 to 1.

**x** numerical; explanatory variable.

## Examples

```
## Not run:
library(zoib)
data("BiRepeated", package = "zoib")
eg2 <- zoib(y1|y2 ~ x|1|x, data= BiRepeated, random=1,n.response=2,
            EUID= BiRepeated$id, joint=TRUE,zero.inflation = FALSE,
            one.inflation = FALSE, prior.Sigma = "VC.unif",
            n.iter=7000,n.thin=25,n.burn=2000)
coeff <- eg2$coeff
summary(coeff)

### check convergence
traceplot(coeff);
autocorr.plot(coeff);
check.psrf(coeff)

### plot posterior mean of y vs. observed y to check on goodness of fit.
n= nrow(BiRepeated)
ypred1 = rbind(eg2$ypred[[1]][,1:n],eg2$ypred[[2]][,1:n])
ypred2 = rbind(eg2$ypred[[1]][,(n+1):(2*n)],eg2$ypred[[2]][,(n+1):(2*n)])
post.mean1 = apply(ypred1,2,mean);
post.mean2 = apply(ypred2,2,mean);

plot(BiRepeated$y1, post.mean1, xlim=c(0,1),ylim=c(0,1), col='green2',
     pch=2,xlab='Observed y', ylab='Predicted y', main="")
points(BiRepeated$y2,post.mean2,col='purple')
abline(0,1,col='red')
legend(0.1,0.9,col=c('green2','purple'),c("y1","y2"),pch=c(2,1))

## End(Not run)
```

---

check.psrf

---

*Convergence Check for Markov Chain Monte Carlo simulations via  
Potential Scale Reduction Factor*


---

## Description

check.psrf computes and summarizes univariate potential scale reduction factor. It also checks whether the multivariate potential scale reduction factor can be calculated.

## Usage

```
check.psrf(post1 = NULL, post2 = NULL, post3 = NULL, post4 = NULL, post5 = NULL)
```

## Arguments

post1                    an mcmc.list with posterior samples from all Markov chains, or a data frame containing the draws from the 1st Markov Chain.

post2	Monte Carlo Posterior draws (data frame) from the 2nd Markov Chain, if the post1 is a data frame; specify post2 only needed
post3	Monte Carlo Posterior draws (data frame) from the 3rd Markov Chain, if the post1 is a data frame; specify post3 only needed
post4	Monte Carlo Posterior draws (data frame) from the 4th Markov Chain, if the post1 is a data frame; specify post4 only needed
post5	Monte Carlo Posterior draws (data frame) from the 5th Markov Chain, if the post1 is a data frame; specify post5 only needed

Details

The posterior samples from each chain are stored in a data frame, with columns representing parameters from the model, and rows presenting posterior draws on the parameters. If the input post1 is a data frame contains the draws from one chain, then check.psrf can take up to 5 chains though it is not necessary to have 5 chain; but at least 2 chains are necessary.

Value

The function outputs

psrf.s	univariate psrf values and the 95% confidence interval from all model parameters.
psrf.m	multivariate psrf if the covariance matrix of the parameters are positive definite.
psrf.s.summ	the summary of the univariate psrf across parameter.

Author(s)

Fang Liu (fang.liu.131@nd.edu)

Examples

```
## Not run:
post1= data.frame(cbind(rnorm(400,0,1), rbeta(400,2,3)))
post2= data.frame(cbind(rnorm(400,0,1), rbeta(400,2,3)))
check.psrf(post1,post2)

## End(Not run)
```

---

GasolineYield	<i>Gasoline Yields Data</i>
---------------	-----------------------------

---

Description

GasolineYield contains proportion of crude oil converted to gasoline after distillation and fractionation.

**Usage**

```
data(GasolineYield)
```

**Format**

A data frame with 32 observations on the following 3 variables.

`yield` a numeric vector; proportion of crude oil converted to gasoline after distillation and fractionation.

`temp` a numeric vector: temperature (F) at which 100% gasoline has vaporized.

`batch` crude oil batch ID.

**Details**

The Gasoline Yields Data is a subset of the data `GasolineYield` in R package "betareg". Refer to the detailed description in R package "betareg".

**Examples**

```
## Not run:
library(zoib)
data("GasolineYield", package = "zoib")

#####
# fixed effects zoib with
# batch as a 10-level qualitative variable
#####

eg.fixed <- zoib(yield ~ temp + as.factor(batch)| 1,
                data=GasolineYield, joint = FALSE, random = 0,
                EUID = 1:nrow(d), zero.inflation = FALSE,
                one.inflation = FALSE, n.iter = 1100, n.thin = 5,
                n.burn=100)
# yields 400 posterior draws (200 per chain) on the model parameters
coeff <- eg.fixed$coef
summary(coeff)

### check on convergence
traceplot(coeff)
autocorr.plot(coeff)
check.psrff(coeff)

### Design Matrix: Xb, Xd, Xb0, Xb1
eg.fixed$Xb; eg.fixed$Xd; eg.fixed$Xb0; eg.fixed$Xb1

# plot posterior mean of y vs. observed y to check on goodness of fit.
ypred = rbind(eg.fixed$ypred[[1]],eg.fixed$ypred[[2]])
post.mean= apply(ypred,2,mean);
plot(GasolineYield$yield, post.mean, col='blue',pch=2);
abline(0,1,col='red')
```

```
#####
# mixed effects zoib with batch as a random variable
#####
eg.random <- zoib(yield ~ temp | 1 | 1, data=GasolineYield,
                 joint = FALSE, random=1, EUID=GasolineYield$batch,
                 zero.inflation = FALSE, one.inflation = FALSE,
                 n.iter=3200, n.thin=15, n.burn=200)
sample2 <- eg.random$coeff
summary(sample2)

# check convergence on the regression coefficients
traceplot(sample2)
autocorr.plot(sample2)
check.psrfr(sample2)

# plot posterior mean of y vs. observed y to check on goodness of fit.
ypred = rbind(eg.random$ypred[[1]],eg.random$ypred[[2]])
post.mean= apply(ypred,2,mean);
plot(GasolineYield$yield, post.mean, col='blue',pch=2);
abline(0,1,col='red')

## End(Not run)
```

---

paraplot

*visual display of the posterior inferences of the parameters from a zoib model*


---

## Description

paraplot plots the posterior mode, mean, or median with Bayesian credible intervals for the parameters from a zoib model. plot.para can accommodate up to 4 groups (models) of inferences.

## Usage

```
paraplot(para1, para2=NULL, para3=NULL, para4=NULL,
         tickx=NULL, jitter=NULL, pch = 1:4, col=1:4,
         legpos=NULL, legtext, annotate=FALSE)
```

## Arguments

para1	a data frame or matrix with posterior inferences from approach (model) 1.
para2	a data frame or matrix with posterior inferences from approach (model) 4; specify para2 only needed.
para3	a data frame or matrix with posterior inferences from approach (model) 3; specify para3 only needed.
para4	a data frame or matrix with posterior inferences from approach (model) 2; specify para4 only needed.
tickx	customization of the tickx of the x-axis.



jitter	jittering distance between the plotting positions of different groups to enhance plot readability.
pch	point types for different groups.
col	colors for different types.
legpos	the positions of legend in the format of c(x,y).
legtext	contents of the legtext, such as the names of the approaches.
annotate	whether to label parameters.

### Details

Each of the applicabale para objects contains 3 columns. The 1st column is the point estimate (posterior mean, median or mode), the 2nd and 3rd columns contains the lower and upper bounds of the Bayesian credible intervals. The rows represent the parameters from an approach (model). It is not necessary that all para objects have the same set of parameters.

### Author(s)

Fang Liu (fang.liu.131@nd.edu)

### Examples

```
## Not run:
set.seed(12)
x=rnorm(4); para1 = cbind(x, x-1,x+1); rownames(para1) = c("a","b","c","d")
x=rnorm(4)+1; para2 = cbind(x, x-1,x+1); rownames(para2) = c("a","b","e","f")
x=rnorm(3)+2; para3 = cbind(x, x-1,x+1); rownames(para3) = c("a","b","d")
paraplot(para1, para2, para3, para4=NULL, legpos=c(-1.5,6),
          legtext=c("model 1","model 2","model 3"),annotate=TRUE)
paraplot(para1, legpos=c(-2,3), legtext="m1", annotate=TRUE)

## End(Not run)
```

---

pred.zoib

*posterior predictive samples of Y for given new X*

---

### Description

generate posterior predictive samples of Y for a given set of new X, and produce posterior summary on the posterior predictive samples

### Usage

```
pred.zoib(object, xnew, summary=TRUE)
```

**Arguments**

object	the object output from function zoib
xnew	a set of new X at each the posterior predictive values are calculated. xnew should be of the same type and format as the X in the original data where the zoib model is fitted
summary	if TRUE (the default), a basic summary on each posterior predictive value, including mean, SD, min, max, med, 2.5% and 97.5% quantiles, are provided.

**Details**

xnew should be in the format of data.frame and should of the same type and have the same variables as the X in the original data where the zoib model is fitted. See the example below.

**Author(s)**

Fang Liu (fang.liu.131@nd.edu)

**Examples**

```
## Not run:
data("GasolineYield")
eg1 <- zoib(yield ~ temp + as.factor(batch)| 1, data=GasolineYield,
           joint = FALSE, random = 0, EUID = 1:nrow(d),
           zero.inflation = FALSE, one.inflation = FALSE,
           n.iter = 1600, n.thin = 2, n.burn=100, seeds=c(1,2),n.chain=2)
xnew <- data.frame(temp = c(205, 218), batch = factor(c(1, 2), levels = 1:10))
ypred <- pred.zoib(eg1, xnew)

data("BiRepeated")
eg2 <- zoib(y1|y2 ~ x|1|x, data= BiRepeated, n.response=2,
           random=1, EUID= BiRepeated$id,
           zero.inflation = FALSE, one.inflation = FALSE,
           prior.Sigma = "VC.unif", n.iter=2100, n.thin=10, n.burn=100)
xnew<- data.frame(x=BiRepeated[1:6,4])
pred.zoib(eg2,xnew)

## End(Not run)
```

---

zoib

---

*Bayesian Inference for Zero/One Inflated Beta Regression*


---

**Description**

Description: zoib fits a zero/one inflated beta regression model and obtains the Bayesian Inference for the model via the Markov Chain Monte Carlo approach implemented in JAGS.

**Usage**

```
zoib(model, data, n.response=1, joint = TRUE,
      zero.inflation = TRUE, one.inflation = TRUE,
      random = 0, EUID, link.mu = "logit", link.x0 = "logit",
      link.x1 = "logit", prec.int = matrix(1e-3, n.response, 4),
      prior.beta = matrix("DN", n.response, 4),
      prec.DN = matrix(1e-3, n.response, 4),
      lambda.L2 = matrix(1e-3, n.response, 4),
      lambda.L1 = matrix(1e-3, n.response, 4),
      lambda.ARD = matrix(1e-3, n.response, 4), prior.Sigma = "VC.unif",
      scale.unif = 20, scale.halfcauchy = 20,
      n.chain = 2, n.iter = 5000, n.burn = 200, n.thin = 2, inits=NULL, seeds=NULL)
```

**Arguments**

model	Symbolic description of the model in the format of formula, such as $y \sim x$ , $y_1 \mid y_2 \sim x_1 + x_2$ , or $y_1 \sim x \mid z$ . Refer to "details" for more information.
data	Data to be analyzed.
n.response	Number of response variables. Default is 1.
joint	Whether to jointly model multiple responses if $n.response \geq 2$ . Default is TRUE.
zero.inflation	A vector that contains $n.response$ values of TRUE or FALSE on whether each of the response variables has inflation at zero. Default is TRUE.
one.inflation	A vector that contains $n.response$ values of TRUE or FALSE on whether each of the response variables has inflation at one. Default is TRUE.
random	Sepecifys which linear predictor(s) or link function(s) contain a random component. Default is 0 (no random component). Refer to "details" for more information.
EUID	Listing of the experimental unit ID in each row of the data set.
link.mu	Link function for the mean of the beta distribution piece of the zoib model. Choices are "logit" (default), "probit" and "cloglog".
link.x0	Link function for $\Pr(y=0)$ . Choices are "logit" (default), "probit" and "cloglog".
link.x1	Link function for $\Pr(y=1 \mid y>0)$ . Choices are "logit" (default), "probit" and "cloglog".
prec.int	Precision parameter of the prior distribution (diffuse normal) of the intercept in the linear predictor of each link function. Default is 0.001 in all 4 link functions for all response variables.
prior.beta	Prior choice for the regression coefficients other than the intercepts in each of the 4 link functions. Default is "diffuse normal" in all 4 link functions for all response variables. Refer to "details" for more information.
prec.DN	Precision parameters of the normal distribution if the diffuse normal prior is chosen as the prior distributions of the regression coefficients in all 4 linear predictors for all response variables. Default precision is 0.001.
lambda.L1	Scale parameter of the prior distributions of the regression coefficients in the linear predictors for all response variables if the L1-like prior is chosen. Refer to the Liu and Kong (2015) and Bae and Mallick (2004) for details.

<code>lambda.L2</code>	Scale parameter of the prior distributions of the regression coefficients in the linear predictors for all response variables if the L2-like prior is chosen. Refer to the Liu and Kong (2015) and Bae and Mallick (2004) for details.
<code>lambda.ARD</code>	Scale parameter in the prior distributions of the regression coefficients in the linear predictors for all response variables if the ARD prior is chosen. Refer to the Liu and Kong (2015) and Bae and Mallick (2004) for details.
<code>prior.Sigma</code>	Prior choice for the variance or the variance-covariance in the case of a single random variable and multiple random variables, respectively. The default is "VC.unif". When there is a single random variable, choose from "VC.unif" and "VC.halfcauchy". Refer to "details" for more information.
<code>scale.unif</code>	Upper bound of the uniform prior for the standard deviation of each random variable if <code>prior.Sigma="VC.unif"</code> is specified. Default is 20.
<code>scale.halfcauchy</code>	Scale parameter of the half-Cauchy prior for the standard deviation of each random variable if <code>prior.Sigma="VC.halfCauchy"</code> is specified. Default is 20.
<code>n.chain</code>	Number of Markov chains from which posterior samples will be drawn ( $\geq 2$ ; default = 2).
<code>n.iter</code>	Number of iterations per chain in the MCMC sampling (default = 5000) before burning-in and thinning.
<code>n.burn</code>	Burning in period of the MCMC chains (default = 200).
<code>n.thin</code>	Thinning period of the MCMC chains after the burn-in (default = 5).
<code>inits</code>	optional specification of initial values for regression coefficients and variance/covariance parameters in the form of a list (see initialization below). If omitted, initial values will be generated automatically. Refer to "details" for more information.
<code>seeds</code>	a vector of dimension <code>n.chain</code> that contains seeds for the initial values and the random number generators of the MCMC chains, if users wish to make the output from the model reproducible.

## Details

\*\*\*\*\* model \*\*\*\*\*

When there are multiple responses  $y$ , each  $y$  should be separated by "|", such as " $y_1 \mid y_2 \mid y_3$ " on the left hand side of the formula. On the right side of the formula, it can include up to 5 parts in the following order:  $x_b \mid x_d \mid x_0 \mid x_1 \mid z$ , where  $x_b$  represents the fixed-effects covariates/factors in the link function of the mean of the beta distribution,  $x_d$  represents the fixed-effects covariates/factors in the link function of the sum of the two shape parameters of the beta distribution,  $x_0$  represents the fixed-effect covariates/factors in the link function of  $\Pr(y=0)$ ,  $x_1$  represents the fixed-effect covariates/factors in the link function of  $\Pr(y=1|y>0)$ , and  $z$  represents the random-effect covariates/factors. The current version of the package only accomodates  $z$  being the same across all the link functions that have a random component.  $x_b$  and  $x_d$  should always be specified, even if  $x_d$  contains only an intercept. If there is no zero inflation in any of the  $y$ 's, then the  $x_0$  part can be omitted, and zoib automatically adjusts the order of the rest of the  $X$  specifications; similarly for the  $x_1$  part and the random effects part  $z$ .

For example, if there are 3 response variables and 2 independent variables ( $xx_1, xx_2$ ), and none of the  $y$ 's have zero inflation, and all have one inflation, then specification of  $y_1 \mid y_2 \mid y_3 \sim xx_1 + xx_2$

$1 \mid xx1 \mid xx2$ , together with  $\text{zero.inflation} = c(F,F,F)$  and  $\text{one.inflation} = c(T,T,T)$ , implies  $xb = (1, xx1, xx2)$ ,  $xd = 1$  (intercept),  $x0 = \text{NULL}$ ,  $x1 = (1, xx1)$  for all  $y$ 's,  $z = (1, xx2)$ . If  $y1$  has inflation at 0, and  $y3$  has inflation at 1, and there is no random effect, model  $y1 \mid y2 \mid y3 \sim xx1 + xx2 \mid xx1 \mid xx1 \mid xx2$ , together with  $\text{zero.inflation} = c(T,F,F)$  and  $\text{one.inflation} = c(F,F,T)$ , implies  $xb = (1, xx1, xx2)$ ,  $xd = (1, xx1)$ ,  $x0 = (1, xx1)$  for  $y1$ ,  $x1 = (1, xx2)$  for  $y3$ . Refer to [Formula](#) in package [Formula](#) for more details on the specification of terms in formula such as interaction terms and categorical  $x$ 's, etc.

\*\*\*\*\* random \*\*\*\*\*

Denote the four link functions by

Eqn 1.  $g(\mu) = xb*b$ , where  $g$  is the link function (logit, probit or cloglog), and  $\mu$  is the mean of the Beta distribution

Eqn 2.  $\log(\eta) = xd*d$ , where  $\eta$  is the sum of the two shape parameters of the Beta distribution

Eqn 3.  $g(p0) = x0*b0$ , where  $g$  is the link function (logit, probit or cloglog), and  $p0 = \Pr(y=0)$

Eqn 4.  $g(p1) = x1*b1$ , where  $g$  is the link function (logit, probit or cloglog), and  $p1 = \Pr(y=1 \mid y>0)$   
then random =

0: no random effect (default);

1: only the linear predictor in Eqn 1 has a random component; that is,  $g(\mu) = xb*b + z*\gamma$

2: only the linear predictor in Eqn 2 has a random component; that is,  $\log(\eta) = xd*d + z*\gamma$

3: only the linear predictor in Eqn 3 has a random component; that is,  $g(p0) = x0*b0 + z*\gamma$

4: only the linear predictor in Eqn 4 has a random component; that is,  $g(p1) = x1*b1 + z*\gamma$

12: the linear predictors in Eqns 1 and 2 contain random component  $z*\gamma$

13: the linear predictors in Eqns 1 and 3 contain random component  $z*\gamma$

14: the linear predictors in Eqns 1 and 4 contain random component  $z*\gamma$

23: the linear predictors in Eqns 2 and 3 contain random component  $z*\gamma$

24: the linear predictors in Eqns 2 and 4 contain random component  $z*\gamma$

34: the linear predictors in Eqns 3 and 4 contain random component  $z*\gamma$

123: the linear predictors in Eqns 1, 2 and 3 contain random component  $z*\gamma$

134: the linear predictors in Eqns 1, 3 and 4 contain random component  $z*\gamma$

124: the linear predictors in Eqn 1, 2 and 4 contain random component  $z*\gamma$

1234: the linear predictors in all Eqns contain random component  $z*\gamma$

\*\*\*\*\* prior.beta \*\*\*\*\*

1. "DN": Diffuse Normal

2. "L1": L1-like shrinkage prior

3. "L2": L2-like shrinkage prior

4. "ARD": ARD-like shrinkage prior

\*\*\*\*\* prior.Sigma \*\*\*\*\*

1. "VC.unif": Sigma is diagonal, the prior for the standard deviation of each random variable follows a uniform distribution;

2. "VC.halfcauchy": Sigma is diagonal, the prior for the standard deviation of each random variable follows a half-Cauchy distribution with a large scale parameter;

3. "UN.unif": Sigma is fully parameterized, the prior for the standard deviation of each random variable follows a uniform distribution; each element in the correlation matrix follows an independent  $\text{unif}(0,1)$  distribution with necessary constraints to ensure positive definiteness.

4. "UN.halfcauchy". Sigma is fully parameterized, the prior for the standard deviation of each random variable follows an half-Cauchy distribution; each element in the correlation matrix follows an independent  $\text{unif}(0,1)$  distribution with necessary constraints to ensure positive definiteness.

\*\*\*\*\* inits \*\*\*\*\*

If supplied, should be in the following parameter order:

`inits = list(list(b=, d=, b0=, b1=, sigma=, R=), list(b=, d=, b0=, b1=, sigma=, R=), ...)`

The notations `b`, `d`, `b0`, `b1` are the same as in the "random" section above, with each specified in a matrix format of dimension  $(n_x+1)*n_{\text{response}}$ , where  $n_x$  is number of regression coefficients corresponding to the respective  $x$ 's (+1 because of the intercept). `sigma` is a vector containing the standard deviation of each random component, and `R` contains the lower triangle of the correlation matrix. For example, in a 3x3 correlation matrix, `R` is specified as `c(1,r21,l,r31,r32,1)`. Each inner list contains the starting values for one MCMC chain. If initial values are specified only for a subset of the parameters in a model, `NULL` should be used for the rest of the unspecified parameters (whose initial values will be generated automatically by the function).

## Value

<code>model</code>	the zoib model
<code>MCMC.model</code>	the MCMC (JAGS) model object, from which samples can be drawn and DIC can be calculated
<code>coeff</code>	posterior samples of regression coefficients from the zoib model
<code>Xb</code>	design matrix in the link function for modeling the mean of the beta regression
<code>Xd</code>	design matrix in the link function for modeling the sum of the two parameters of the beta regression
<code>X0</code>	design matrix in the link function for modeling $\Pr(y=0)$
<code>X1</code>	design matrix in the link function for modeling $\Pr(y=1 y>0)$
<code>yobs</code>	observed $Y$
<code>ypred</code>	posterior predictive samples of $Y$
<code>resid</code>	residual
<code>resid.std</code>	standardized residual

## Author(s)

Fang Liu (<fang.liu.131@nd.edu>)

## References

- Liu, F. and Kong, Y. (2015). ZOIB: an R Package for Bayesian Inferences in Beta and Zero One Inflated Beta Regression Models, *The R Journal*, 7(2):34-51
- Liu, F. and Li, Q. (2014) A Bayesian Model for Joint Analysis of Multivariate Repeated Measures and Time to Event Data in Crossover Trials, *Statistical Methods in Medical Research*, doi: 10.1177/0962280213519594
- Liu, F. and Eugenio, E.(2016). A review and comparison of Bayesian and likelihood-based inferences in beta regression and zero-or-one-inflated beta regression, *Statistical Methods in Medical Research*, DOI: 10.1177/0962280216650699

**Examples**

```
## Not run:  
#refer to data sets GasolineYield, BiRepeated, and AlcoholUse in package zoib  
#for examples on fixed effect models, mixed effects models, joint modeling  
#of bivariate beta variables with repeated measures, and modelling clustered  
#beta variables with inflation at 0 using zoib  
  
## End(Not run)
```

# Index

## \* **datasets**

- AlcoholUse, [3](#)
- BiRepeated, [4](#)
- GasolineYield, [6](#)

## \* **package**

- zoib-package, [2](#)

AlcoholUse, [3](#)

BiRepeated, [4](#)

check.psr, [5](#)

Formula, [13](#)

GasolineYield, [6](#)

paraplot, [8](#)

pred.zoib, [9](#)

zoib, [10](#)

zoib-package, [2](#)