Supplementary Material for
"A Review and Comparison of Bayesian and Likelihood-Based Inferences in Beta Regression and Zero-or-One Inflated Beta Regressio"

Fang Liu* and Evercita C. Eugenio

Department of Applied and Computational Mathematics and Statistics

University of Notre Dame, Notre Dame, IN 46556

*fang.liu.131@nd.edu

S1: An example on the posterior distribution of regression coefficients and covariance matrix in beta regression and zoib regression

If the following priors on $\boldsymbol{\beta}$ and $\Sigma = \mathrm{diag}(\sigma_l)\mathbf{R}\mathrm{diag}(\sigma_l)$ are applied,

$$\begin{aligned}
f(\boldsymbol{\beta}) &\propto \textstyle\prod_{j=1}^{p}\prod_{m=1}^{4}\prod_{k=0}^{K_{mj}} \exp(-\beta_{m,jk}^2\tau^2/2) \text{ (independent DN)}\\
f(\sigma_1,\ldots,\sigma_L) &\propto \textstyle\prod_{l=1}^{L}(1+(\sigma_l/C)^2)^{-1} \text{ (half-Cauchy distribution)}\\
f(\mathbf{R}) &\propto \text{constant (joint uniform distribution on correlation matrix } \mathbf{R})
\end{aligned}$$

then we can obtain the following conditional posterior distribution

$$\begin{aligned}
f(\boldsymbol{\gamma}|\mathbf{y},\boldsymbol{\beta},\Sigma) \propto \prod_{i=1}^{n}\prod_{j=1}^{p}\Bigg\{ & \left(\frac{\exp(\mathbf{x}_{3,ij}\boldsymbol{\beta}_{3,j}+\mathbf{z}_{ij}\boldsymbol{\gamma}_i)}{1+\exp(\mathbf{x}_{3,ij}\boldsymbol{\beta}_{3,j}+\mathbf{z}_{ij}\boldsymbol{\gamma}_i)}\right)^{\mathrm{I}(y_{ij}=0)} \left(\frac{1}{1+\exp(\mathbf{x}_{3,ij}\boldsymbol{\beta}_{3,j}+\mathbf{z}_{ij}\boldsymbol{\gamma}_i)}\right)^{\mathrm{I}(y_{ij}>0)}\\
&\times\left(\frac{\exp(\mathbf{x}_{4,ij}\boldsymbol{\beta}_{4,j}+\mathbf{z}_{ij}\boldsymbol{\gamma}_i)}{1+\exp(\mathbf{x}_{4,ij}\boldsymbol{\beta}_{4,j}+\mathbf{z}_{ij}\boldsymbol{\gamma}_i)}\right)^{\mathrm{I}(y_{ij}=1)}\\
&\times\Bigg\{\frac{\Gamma(\exp(\mathbf{x}_{2,ij}\boldsymbol{\beta}_{2,j}+\mathbf{z}_{ij}\boldsymbol{\gamma}_i))}{\Gamma\left(\frac{\exp(\mathbf{x}_{2,ij}\boldsymbol{\beta}_{2,j}+\mathbf{x}_{1,ij}\boldsymbol{\beta}_{1,j}+2\mathbf{z}_{ij}\boldsymbol{\gamma}_i)}{1+\exp(\mathbf{x}_{1,ij}\boldsymbol{\beta}_{1,j}+\mathbf{z}_{ij}\boldsymbol{\gamma}_i)}\right)\Gamma\left(\frac{\exp(\mathbf{x}_{2,ij}\boldsymbol{\beta}_{2,j}+\mathbf{z}_{ij}\boldsymbol{\gamma}_i)}{1+\exp(\mathbf{x}_{1,ij}\boldsymbol{\beta}_{1,j}+\mathbf{z}_{ij}\boldsymbol{\gamma}_i)}\right)}\\
&\quad (y_{ij})^{\frac{\exp(\mathbf{x}_{2,ij}\boldsymbol{\beta}_{2,j}+\mathbf{x}_{1,ij}\boldsymbol{\beta}_{1,j}+2\mathbf{z}_{ij}\boldsymbol{\gamma}_i)}{1+\exp(\mathbf{x}_{1,ij}\boldsymbol{\beta}_{1,j}+\mathbf{z}_{ij}\boldsymbol{\gamma}_i)}-1}(1-y_{ij})^{\frac{\exp(\mathbf{x}_{2,ij}\boldsymbol{\beta}_{2,j}+\mathbf{z}_{ij}\boldsymbol{\gamma}_i)}{1+\exp(\mathbf{x}_{1,ij}\boldsymbol{\beta}_{1,j}+\mathbf{z}_{ij}\boldsymbol{\gamma}_i)}-1}\Bigg\}^{\mathrm{I}(y_{ij}\in(0,1))}\Bigg\}\\
\times |\Sigma|^{-1/2}\exp(-\boldsymbol{\gamma}^t\Sigma^{-1}\boldsymbol{\gamma}/2)&\\
f(\boldsymbol{\beta}|\mathbf{y},\boldsymbol{\gamma},\Sigma) \propto \prod_{i=1}^{n}\prod_{j=1}^{p}\Bigg\{ & \left(\frac{\exp(\mathbf{x}_{3,ij}\boldsymbol{\beta}_{3,j}+\mathbf{z}_{ij}\boldsymbol{\gamma}_i)}{1+\exp(\mathbf{x}_{3,ij}\boldsymbol{\beta}_{3,j}+\mathbf{z}_{ij}\boldsymbol{\gamma}_i)}\right)^{\mathrm{I}(y_{ij}=0)} \left(\frac{1}{1+\exp(\mathbf{x}_{3,ij}\boldsymbol{\beta}_{3,j}+\mathbf{z}_{ij}\boldsymbol{\gamma}_i)}\right)^{\mathrm{I}(y_{ij}>0)}\\
&\times\left(\frac{\exp(\mathbf{x}_{4,ij}\boldsymbol{\beta}_{4,j}+\mathbf{z}_{ij}\boldsymbol{\gamma}_i)}{1+\exp(\mathbf{x}_{4,ij}\boldsymbol{\beta}_{4,j}+\mathbf{z}_{ij}\boldsymbol{\gamma}_i)}\right)^{\mathrm{I}(y_{ij}=1)}\\
&\times\Bigg\{\frac{\Gamma(\exp(\mathbf{x}_{2,ij}\boldsymbol{\beta}_{2,j}+\mathbf{z}_{ij}\boldsymbol{\gamma}_i))}{\Gamma\left(\frac{\exp(\mathbf{x}_{2,ij}\boldsymbol{\beta}_{2,j}+\mathbf{x}_{1,ij}\boldsymbol{\beta}_{1,j}+2\mathbf{z}_{ij}\boldsymbol{\gamma}_i)}{1+\exp(\mathbf{x}_{1,ij}\boldsymbol{\beta}_{1,j}+\mathbf{z}_{ij}\boldsymbol{\gamma}_i)}\right)\Gamma\left(\frac{\exp(\mathbf{x}_{2,ij}\boldsymbol{\beta}_{2,j}+\mathbf{z}_{ij}\boldsymbol{\gamma}_i)}{1+\exp(\mathbf{x}_{1,ij}\boldsymbol{\beta}_{1,j}+\mathbf{z}_{ij}\boldsymbol{\gamma}_i)}\right)}\\
&\quad (y_{ij})^{\frac{\exp(\mathbf{x}_{2,ij}\boldsymbol{\beta}_{2,j}+\mathbf{x}_{1,ij}\boldsymbol{\beta}_{1,j}+2\mathbf{z}_{ij}\boldsymbol{\gamma}_i)}{1+\exp(\mathbf{x}_{1,ij}\boldsymbol{\beta}_{1,j}+\mathbf{z}_{ij}\boldsymbol{\gamma}_i)}-1}(1-y_{ij})^{\frac{\exp(\mathbf{x}_{2,ij}\boldsymbol{\beta}_{2,j}+\mathbf{z}_{ij}\boldsymbol{\gamma}_i)}{1+\exp(\mathbf{x}_{1,ij}\boldsymbol{\beta}_{1,j}+\mathbf{z}_{ij}\boldsymbol{\gamma}_i)}-1}\Bigg\}^{\mathrm{I}(y_{ij}\in(0,1))}\Bigg\}\\
\times \prod_{j=1}^{p}\prod_{m=1}^{4}\prod_{k=0}^{K_{mj}}\left\{\exp(-\beta_{m,jk}^2\tau^2/2)\right\}&\\
f(\Sigma|\mathbf{y},\boldsymbol{\beta},\boldsymbol{\gamma}) \propto |\Sigma|^{-1/2}\exp(-\boldsymbol{\gamma}^t\Sigma^{-1}\boldsymbol{\gamma}/2)&\prod_{l=1}^{L}\left(1+(\sigma_l/C)^2\right)^{-1}
\end{aligned}$$

S2: Tables in the second simulation study (zero/one inflated data)

Table 1: Parameter values in Simulation 2

y-separation	inflation (%)	$\beta_{1,0}$	$\beta_{1,1}$	$\beta_{2,0}$	$\beta_{3,0}$	$\beta_{3,1}$	$\beta_{4,0}$	$\beta_{4,1}$
none	5	0.5	1.5	1.0	-3.37	1.0	-4.20	2.0
	10	0.5	1.5	1.0	-2.56	1.0	-3.00	2.0
	20	0.5	1.5	1.0	-1.65	1.0	-1.38	2.0
at 0	5	2.0	0.5	2.5	-3.37	1.0	-4.20	2.0
	10	2.0	0.5	2.5	-2.56	1.0	-3.00	2.0
	20	2.0	0.5	2.5	-1.65	1.0	-1.38	2.0
at 1	5	-1.0	0.2	2.5	-3.37	1.0	-4.20	2.0
	10	-1.0	0.2	2.5	-2.56	1.0	-3.00	2.0
	20	-1.0	0.2	2.5	-1.65	1.0	-1.38	2.0
at 0, 1	5	-0.1	0.1	2.5	-3.37	1.0	-4.20	2.0
	10	-0.1	0.1	2.5	-2.56	1.0	-3.00	2.0
	20	-0.1	0.1	2.5	-1.65	1.0	-1.38	2.0

Table 2: Percentage of "usable" data sets in the MLE approach

y-separation	0, 1			0			1			none		
inflation (%)	5	10	20	5	10	20	5	10	20	5	10	20
30	91.4	95.8	99.1	88.8	94.5	98.3	94.4	97.0	98.4	95.6	96.3	98.6
n = 100	99.5	100	100	99.9	100	100	99.7	99.9	100	99.5	100	100
300	100	100	100	100	100	100	100	100	100	100	100	100

S3: Additional results from simulation study 3 (beta regression with repeated measures)

Table 3: Results in simulation study 3 (likelihood-based approach via PROC NLMIXED; starting values were within ±20% ∼ 30% of the true parameter values)

ρ	parameter	true value	n=100 Bias	n=100 RMSE	n=100 CP(%)	n=100 M[a]	n=300 Bias	n=300 RMSE	n=300 CP(%)	n=300 M[a]
	β_{01}	-1.0	-0.795	0.954	25.6		-1.394	1.409	0.9	
	β_{11}	-2.0	1.733	1.934	17.9		2.107	2.117	0.9	
	ν_1	2.5	-0.586	0.682	17.7		-1.541	1.553	0.9	
	β_{02}	1.0	-0.292	0.376	43.3		-0.156	0.229	40.6	
0.1	β_{12}	2.0	0.150	0.182	95.0	310	0.266	0.272	34.0	456
	ν_2	3.0	-0.491	0.566	17.3		-0.507	0.518	0.9	
	σ_1^2	0.2	0.956	1.346	53.8		1.697	1.830	13.8	
	σ_2^2	0.2	0.524	0.584	67.6		0.748	0.757	5.0	
	σ_{12}^2	0.1	-0.904	1.006	16.1		-1.090	1.095	0.9	
	β_{01}	-1.0	-0.591	0.777	37.8		-0.436	0.498	17.7	
	β_{11}	-2.0	1.542	1.873	30.1		2.217	2.330	9.1	
	ν_1	2.5	-0.902	1.108	30.3		-2.829	3.065	8.4	
	β_{02}	1.0	-0.372	0.496	46.1		-0.280	0.334	25.8	
-0.5	β_{12}	2.0	0.380	0.444	42.4	353	0.491	0.511	7.6	394
	ν_2	3.0	-0.371	0.472	28.1		-0.792	0.936	8.1	
	σ_1^2	0.2	0.245	0.645	84.6		0.253	0.576	72.7	
	σ_2^2	0.2	0.578	0.693	58.2		0.860	0.905	6.0	
	σ_{12}^2	-0.5	-0.405	0.449	25.0		-0.473	0.489	4.8	
	β_{01}	-1.0	-0.338	0.761	68.5		-0.556	0.960	55.4	
	β_{11}	-2.0	0.945	1.473	55.2		0.692	1.072	49.3	
	ν_1	2.5	-0.796	1.486	54.7		-0.220	0.357	45.6	
	β_{02}	1.0	-0.166	0.526	60.4		-0.317	0.570	55.4	
0.8	β_{12}	2.0	0.053	0.416	54.8	223	-0.140	0.252	58.4	159
	ν_2	3.0	-0.220	0.414	60.1		-0.160	0.305	46.8	
	σ_1^2	0.2	0.173	0.317	63.5		1.009	1.937	65.8	
	σ_2^2	0.2	0.082	0.180	83.3		0.193	0.294	74.2	
	σ_{12}^2	0.8	-0.437	0.648	52.8		-0.288	0.604	78.3	

[a] all data with finite lower and upper CI bounds were used in the calculation despite warning in non-positive definite Hessian matrices

Table 4: Simulation results on the likelihood-based approach via PROC NLMIXED in the bivariate response with repeated measures simulation (starting values were set at the true parameter values)

ρ	parameter	true	n=100				n=300			
			Bias	RMSE	CP(%)	M[a]	Bias	RMSE	CP(%)	M[a]
	β_{01}	-1.0	-0.004	0.062	96.2		0.001	0.037	95.0	
	β_{11}	-2.0	0.001	0.170	95.7		0.008	0.099	95.2	
	ν_1	2.5	0.012	0.060	97.1		0.007	0.035	94.5	
	β_{02}	1.0	-0.001	0.058	96.7		0.002	0.033	96.1	
0.1	β_{12}	2.0	-0.010	0.131	97.1	209	0.004	0.079	95.6	460
	ν_2	3.0	0.006	0.060	97.6		0.006	0.039	94.1	
	σ_1^2	0.2	0.005	0.0383	97.6		0.002	0.025	94.1	
	σ_2^2	0.2	-0.024	0.061	97.6		-0.006	0.036	96.7	
	σ_{12}^2	0.1	0.042	0.162	99.0		-0.002	0.098	98.5	
	β_{01}	-1.0	0.005	0.066	94.2		-0.002	0.037	95.2	
	β_{11}	-2.0	-0.005	0.181	94.2		0.014	0.106	94.2	
	ν_1	2.5	0.013	0.068	92.8		0.006	0.035	94.2	
	β_{02}	1.0	0.009	0.060	95.2		-0.000	0.035	93.9	
-0.5	β_{12}	2.0	0.001	0.151	92.3	208	0.009	0.089	91.1	293
	ν_2	3.0	0.009	0.067	92.3		0.003	0.035	95.6	
	σ_1^2	0.2	0.002	0.041	95.7		-0.003	0.024	94.2	
	σ_2^2	0.2	-0.007	0.064	98.6		0.004	0.041	93.9	
	σ_{12}^2	-0.5	0.045	0.157	98.1		-0.001	0.102	97.3	
	β_{01}	-1.0	0.006	0.066	94.5		-0.004	0.038	95.9	
	β_{11}	-2.0	-0.003	0.148	99.2		0.022	0.104	93.5	
	ν_1	2.5	0.010	0.056	95.3		0.009	0.035	96.2	
	β_{02}	1.0	0.002	0.059	96.1		0.001	0.033	95.6	
0.8	β_{12}	2.0	-0.000	0.152	92.9	127	-0.000	0.081	95.6	339
	ν_2	3.0	0.020	0.070	92.9		0.008	0.037	94.7	
	σ_1^2	0.2	0.014	0.047	94.5		0.004	0.024	95.6	
	σ_2^2	0.2	-0.033	0.060	98.4		-0.016	0.034	96.5	
	σ_{12}^2	0.8	0.089	0.154	99.2		0.033	0.093	98.2	

[a] The number of usable simulations repetitions out of 500 after removing those whose final Hessian matrix was not positive definite

S4: core SAS and R codes used in the simulation studies and the case study

S4.1 SAS codes

```
********************* Simulation 2 ********************;
* zero inflation *;
proc nlmixed data = zero TECH=QUANEW;
by sim;
parms a0=1 b0=-1 a=0 b=0 d=2.5;
pizero = exp(a0+b0*x)/(1 + exp(a0+b0*x));
mu = exp(a+b*x)/(1 + exp(a+b*x));
phi = exp(d);
w = mu*phi;
t = (phi-mu)*phi;
if (y = 0) then ll = log(pizero);
else ll = lgamma(w+t)-lgamma(w)-lgamma(t)+
          ((w-1)*log(y))+((t-1)*log(1-y))+log(1-pizero);
model y ~ general(ll);
predict mu out=mu_results0 (keep=y pred);
predict phi out=phi_results0 (keep=y pred);
predict pizero out=pizero_results0 (keep=y pred);
run;
data prediction;
merge zero mu_results0 phi_results0 pizero_results0;
run;

* one inflation *;
proc nlmixed data = one TECH=QUANEW;
by sim;
parms a1=1 b1=-1 a=0 b=0 d=2.5;
pione = exp(a1+b1*x)/(1 + exp(a1+b1*x));
mu = exp(a+b*x)/(1 + exp(a+b*x));
phi = exp(d);
w = mu*phi;
t = phi-mu*phi;
if (y = 1) then ll = log(pione);
else ll = lgamma(w+t)-lgamma(w)-lgamma(t)+
         ((w-1)*log(y))+((t-1)*log(1-y))+log(1-pione);
model y ~ general(ll);
predict mu out=mu_results1 (keep=y pred);
predict phi out=phi_results1 (keep=y pred);
predict pione out=pione_results1 (keep=y pred);
run;
data prediction;
```

```
merge one mu_results1 phi_results1 pione_results1;
run;

* zero one inflation *;
proc nlmixed data = zeroone TECH=QUANEW;
by sim;
parms a0=1 b0=-1 a1=1 b1=-1 a=0 b=0 d=2.5;
pizero = exp(a0+b0*x)/(1 + exp(a0+b0*x));
pione = exp(a1+b1*x)/(1 + exp(a1+b1*x));
mu = exp(a+b*x)/(1 + exp(a+b*x));
phi = exp(d);
w = mu*phi;
t = phi-mu*phi;
if (y = 0) then ll = log(pizero);
else if (y = 1 ) then ll = log(1-pizero) + log(pione);
else ll = lgamma(w+t)-lgamma(w)-lgamma(t)+
         ((w-1)*log(y))+((t-1)*log(1-y))+log(1-pizero)+log(1-pione);
model y ~ general(ll);
predict mu out=mu_results01 (keep=y pred);
predict phi out=phi_results01 (keep=y pred);
predict pizero out=pizero_results01 (keep=y pred);
predict pione out=pione_results01 (keep=y pred);
run;
data prediction;
merge zeroone mu_results01 mu_results01 pizero_results01 pione_results01;
run;

* no inflation *;
proc nlmixed data = none TECH=QUANEW;
by sim;
parms a=0 b=0 d=2.5;
mu = exp(a+b*x)/(1 + exp(a+b*x));
phi = exp(d);
w = mu*phi;
t = phi-mu*phi;
ll = lgamma(w+t)-lgamma(w)-lgamma(t)+((w-1)*log(y))+((t-1)*log(1-y));
model y ~ general(ll);
predict mu out=mu_results (keep=y pred);
predict phi out=phi_results (keep=y pred);
run;
data prediction;
merge none mu_results phi_results;
run;

* ******************** Simulation 3 ********************;
proc nlmixed data=sim3 hessian noad;
```

```
by simnum;

parms beta11=-1.3 beta12=0.7 beta21=-2.6 beta22=2.6
nu1=1.75 nu2=3.9 s21=0.16 rho=0.12 s22=0.25;

s212 = rho*sqrt(s21*s22);
logit1=(beta11+u1)+(beta12+u2)*simx;
alpha11=(exp(logit1)*exp(nu1)) / (1+exp(logit1));
alpha12=exp(nu1) -alpha11;
logit2=(beta21+u1)+(beta22+u2)*simx;
alpha21=(exp(logit2)*exp(nu2)) / (1+exp(logit2));
alpha22=exp(nu2) -alpha21;

fy1=(gamma(alpha11+alpha12)/(gamma(alpha11)*gamma(alpha12))) *
     y**(alpha11-1) * (1-y)**(alpha12-1);
fy2=(gamma(alpha21+alpha22)/(gamma(alpha21)*gamma(alpha22))) *
     y**(alpha21-1) * (1-y)**(alpha22-1);
ll = (which=1)*log(fy1) + (which=2)*log(fy2);

model y ~ general(ll);
random u1 u2 ~ normal([0,0],[s21,s212,s22]) subject=id;
run;
```

S4.2 R codes

```
#-------------------------------------------------------------------------
# Simulation 1
# betareg model
model.betareg=betareg(simy~simx|simx,data=mat[[i]])
# zoib model
model.zoib=zoib(simy~simx|simx,data=mat[[i]],joint=FALSE,random=0,
zero.inflation=FALSE,one.inflation=FALSE,n.iter=1505,n.thin=5,n.burn=0)

#-------------------------------------------------------------------------
# Simulation 2
# betareg model
model.betareg=betareg(simy~simx|1,data=betareg.mat[[i]])

# zoib model
# check to make sure whether data has 0's or 1's, then decide on zoib model
A=(round(zoib.mat[[i]]$y,8)==0);
B=(round(zoib.mat[[i]]$y,8)==1);

tmp = zoib.mat[[i]]
n0 = sum(tmp$y==0)
n1 = sum(tmp$y==1)
```

```
prior.coeff = matrix(rep("DN",4),1,4)
lambda = matrix(0.1,1,4)

if(n0>0) {
    x0 = sort(tmp[tmp$y==0,1])

    x.0 = sort(tmp[tmp$y>0,1], decreasing = TRUE)
    pover01 = 0.5/(n+0.5)
    for(j in 1:(n-n0-1)){
        if(x0[1]<=x.0[j] & x0[1]>=x.0[j+1]){
            pover01 = (2*j+0.5)/(n+0.5); break }}

    x.0 = sort(tmp[tmp$y>0,1])
    pover02 = 0.5/(n+0.5)
    for(j in 1:(n-n0-1)){
        if(x0[n0]>=x.0[j] & x0[n0]<=x.0[j+1]){
            pover02= (2*j+0.5)/(n+0.5); break }}

    pover0=min(pover01, pover02)
    if(pover0<=0.5) {
        prior.coeff[1,3]="L1"; lambda[1,3]=l1*(l1/l2)^(-2*pover0)}
}

if(n1>0) {
    x1 = sort(tmp[tmp$y==1,1])

    x.01 = sort(tmp[tmp$y>0 & tmp$y<1,1],decreasing = TRUE)
    pover11 = 0.5/(n-n0+0.5)
    for(j in 1:(n-n0-n1-1)){
        if(x1[1]<=x.01[j] & x1[1]>=x.01[j+1]){
            pover11 = (2*j+0.5)/(n-n0+0.5); break }}

    x.01 = sort(tmp[tmp$y>0 & tmp$y<1,1])
    pover12 = 0.5/(n-n0+0.5)
    for(j in 1:(n-n0-n1-1)){
        if(x1[n1]>=x.01[j] & x1[n1]<=x.01[j+1]){
            pover12 = (2*j+0.5)/(n-n0+0.5); break }}

    pover1=min(pover11, pover12)
    if(pover1<=0.5){
        prior.coeff[1,4]="L1";  lambda[1,4]=l1*(l1/l2)^(-2*pover1)*(1-inp)}
}

# contains both 0's and 1's
if (any(A) & any(B)) {
    # specify model contains both 0 and 1
```

```
    model.zoib=zoib(y~x|1|x|x,data=zoib.mat[[i]],joint=FALSE,random=0,
    prior.beta=prior.coeff,lambda.L1=lambda,
    zero.inflation=TRUE,one.inflation=TRUE,n.iter=3200,
    n.thin=10,n.burn=100)
}
# contains only 0's
if (any(A) & all(!B)) {
    # specify model contains only 0's
    model.zoib=zoib(y~x|1|x,data=zoib.mat[[i]],joint=FALSE,random=0,
    prior.beta=prior.coeff,lambda.L1=lambda,
    zero.inflation=TRUE,one.inflation=FALSE,n.iter=3200,
    n.thin=10,n.burn=100)
}

# contains only 1's
if (all(!A) & any(B)) {
    # specify model contains only 1's
    model.zoib=zoib(y~x|1|x,data=zoib.mat[[i]],joint=FALSE,random=0,
    prior.beta=prior.coeff,lambda.L1=lambda,
    zero.inflation=FALSE,one.inflation=TRUE,n.iter=3200,
    n.thin=10,n.burn=100)
}

# contains no 0's and 1's
if (all(!A) & all(!B)) {
    # specify model contains both 0 and 1
    model.zoib=zoib(y~x|1,data=zoib.mat[[i]],joint=FALSE,random=0,
    prior.beta=prior.coeff,lambda.L1=lambda,
    zero.inflation=FALSE,one.inflation=FALSE,n.iter=3200,
    n.thin=10,n.burn=100)
}

#------------------------------------------------------------------------
# Simulation 3

# betareg model
model.betareg=betareg(simy1~simx|1,data=mat[[i]])

# zoib model
model.zoib=zoib(simy1|simy2~simx|1|simx,data=mat[[i]],n.response=2,
zero.inflation=FALSE,one.inflation=FALSE,joint=TRUE,random=1,
EUID=mat[[i]]$id,prior.Sigma="UN.halfcauchy",n.chain=2,
n.iter=15200,n.burn=200,n.thin=100,
inits=list(list(b0=NULL,b1=NULL,b=matrix(c(-1.3,-2.6,0.7,2.6),2,2),
        d=matrix(c(1.75,3.9),1,2),sigma=c(0.16,0.25),R=c(1,0.12,1)),
        list(b0=NULL,b1=NULL,b=matrix(c(-0.7,-1.4,1.3,1.4),2,2),
```

```
        d=matrix(c(3.25,2.1),1,2),sigma=c(0.25,0.16),R=c(1,0.08,1))))
```

S4.3 case study

```
mymode<-function(sample)
{
  den<- density(sample)
  mode<-den$x[order(den$y,decreasing = TRUE)][1]
  return(mode)
}

library(betareg)
library(zoib)
data("ReadingSkills", package = "betareg")

n= nrow(ReadingSkills)
y2 =ReadingSkills[,1]
y1 = (y2*50-0.5)/(50-1)
n1 = sum(y1==1)

library(aplpack)
stem.leaf.backback(y2, y1)

Dx=rep(0,n)
for(i  in 1:n){
  if(ReadingSkills[i,2]=="yes") Dx[i]=1
  else Dx[i]=-1}
iq = ReadingSkills[,3]

# betareg: beta regression on y2
rs <- betareg(y2 ~ Dx * iq | Dx + iq, hessian = TRUE)
summary(rs)$coeff
cov=rs$vcov[1:2,1:2]
meancoeff = summary(rs)$coeff$mean[1:2,1]
tmp1 = meancoeff[1]+meancoeff[2]
var1 = cov[1,1] +  cov[2,2]+ 2*cov[1,2]
tmp0 = meancoeff[1]-meancoeff[2]
var0 = cov[1,1] +  cov[2,2]- 2*cov[1,2]
mean1 = exp(tmp1)/(1+exp(tmp1))
varmean1 = exp(2*tmp1)/(1+exp(tmp1))^4
mean0 = exp(tmp0)/(1+exp(tmp0))
varmean0 = exp(2*tmp0)/(1+exp(tmp0))^4

c(mean0, mean0-1.96*sqrt(varmean0), mean0+1.96*sqrt(varmean0))
c(mean1, mean1-1.96*sqrt(varmean1), mean1+1.96*sqrt(varmean1))
c(mean0-mean1, mean0-mean1-1.96*sqrt(varmean1+varmean0),
```

```
  mean0-mean1+1.96*sqrt(varmean1+varmean0))

# zoib: beta regression
x  <- cbind(Dx, iq, Dx*iq)
eg2.fixed <- zoib(y2 ~  x | Dx + iq,  joint = FALSE, random = 0,
    EUID = 1:n, zero.inflation = FALSE, one.inflation = FALSE,
    n.iter= 4200,n.thin= 20, n.burn=200)
coeff <- eg2.fixed$coeff
post.sample <- rbind(coeff[[1]],coeff[[2]])

tmp1 <- exp(post.sample[,1]+post.sample[,2]); mean1 <- tmp1/(1+tmp1)
tmp2 <- exp(post.sample[,1]-post.sample[,2]);  mean0 <- tmp2/(1+tmp2)
mean(mean1); mymode(mean1); quantile(mean1, c(0.025, 0.975))
mean(mean0); mymode(mean0); quantile(mean0, c(0.025, 0.975))
mean(mean0-mean1); mymode(mean0-mean1); quantile(mean0-mean1, c(0.025, 0.975))

summ = summary(coeff)
apply(post.sample,2,hist)
cbind(apply(post.sample,2, mymode), summ[[1]][,1], summ[[2]][,c(1,5)])

# zoib: 1-inflated beta regression
# check data separation at 0 and 1
summary(x[y1==1,]) # all Dx=-1
summary(x[y1<1,])
eg2.fixed1 <- zoib(y1 ~  x | Dx + iq| iq,joint = FALSE, random = 0, EUID= 1:n,
                  zero.inflation = FALSE,one.inflation = TRUE,
                 # prior.beta = matrix(c("DN","DN","DN","DN"),nrow=1),
                 # lambda.L2 =  matrix(rep(0.05,4),nrow=1),
                  n.iter= 4200,n.thin= 20, n.burn=200)
coeff <- eg2.fixed1$coeff
post.sample <- rbind(coeff[[1]],coeff[[2]])

p1 <-exp(post.sample[,5])/(1+exp(post.sample[,5]))
mean1 <- p1+(1-p1)*
    (exp(post.sample[,1]+post.sample[,2])/(1+exp(post.sample[,1]+post.sample[,2])))
mean0 <- p1+(1-p1)*
    (exp(post.sample[,1]-post.sample[,2])/(1+exp(post.sample[,1]-post.sample[,2])))
mean(mean1); mymode(mean1); quantile(mean1, c(0.025, 0.975))
mean(mean0); mymode(mean0); quantile(mean0, c(0.025, 0.975))
mean(mean0-mean1); mymode(mean0-mean1); quantile(mean0-mean1, c(0.025, 0.975))

summ = summary(coeff)
cbind(apply(post.sample,2, mymode), summ[[1]][,1], summ[[2]][,c(1,5)])
```