

# A beta-binomial mixed-effects model approach for analysing longitudinal discrete and bounded outcomes

## *Supplementary Material*

*Josu Najera-Zuloaga, Dae-Jin Lee and Inmaculada Arostegui*

*July 2018*

## Contents

<b>1</b>	<b>Summary</b>	<b>1</b>
<b>2</b>	<b>Simulation results as boxplots and larger sample sizes</b>	<b>1</b>
<b>3</b>	<b>R code for simulation study</b>	<b>1</b>
<b>4</b>	<b>R code for COPD data analysis</b>	<b>5</b>

## 1 Summary

This document contains supplementary material to the manuscript ‘*A beta-binomial mixed-effects model approach for analysing longitudinal discrete and bounded outcomes*’ by Josu Najera-Zuloaga, Dae-Jin Lee and Inmaculada Arostegui.

Sections 2, includes the additional results to the simulation study performed in the manuscript. For reproducibility of the methods developed in our paper we provide in Section 3, the R code used in the simulation study and in Section 4, the R code for the real data analysis.

All R functions, codes and data are available at <http://idaejin.github.io/software/>

## 2 Simulation results as boxplots and larger sample sizes

Figure 1 allows for us to study the performances of both approaches in terms of the biases and ESD. The boxplots of the estimates of  $\beta_1$  and  $\log(\phi)$  for the different scenarios are displayed. In the left panel, we can observe the resulting estimates for  $\beta_1$ , while in the right panel, the estimates for the dispersion parameter in the log scale ( $\log(\phi)$ ) are shown. As shown in the numerical results in Table 2 (of the main manuscript), as  $\sigma$  increases, the bias of the estimates of both parameters increases. Although the ESD of GAMLSS is lower than *BBmm*, our ESD results are closer to ADS. Moreover, the estimates of GAMLSS have larger biases than the *BBmm* approach.

Next table shows the results of the simulation study for larger samples sizes ( $n = 500$  and  $n = 1000$ ) only for  $\sigma = 1.5$  and  $\phi = 1.5$ .

Figure 2, shows the boxplots of the estimates of  $\beta_1$  and  $\log(\phi)$  for  $n = 500$  and  $n = 1000$ .

## 3 R code for simulation study

Load R libraries

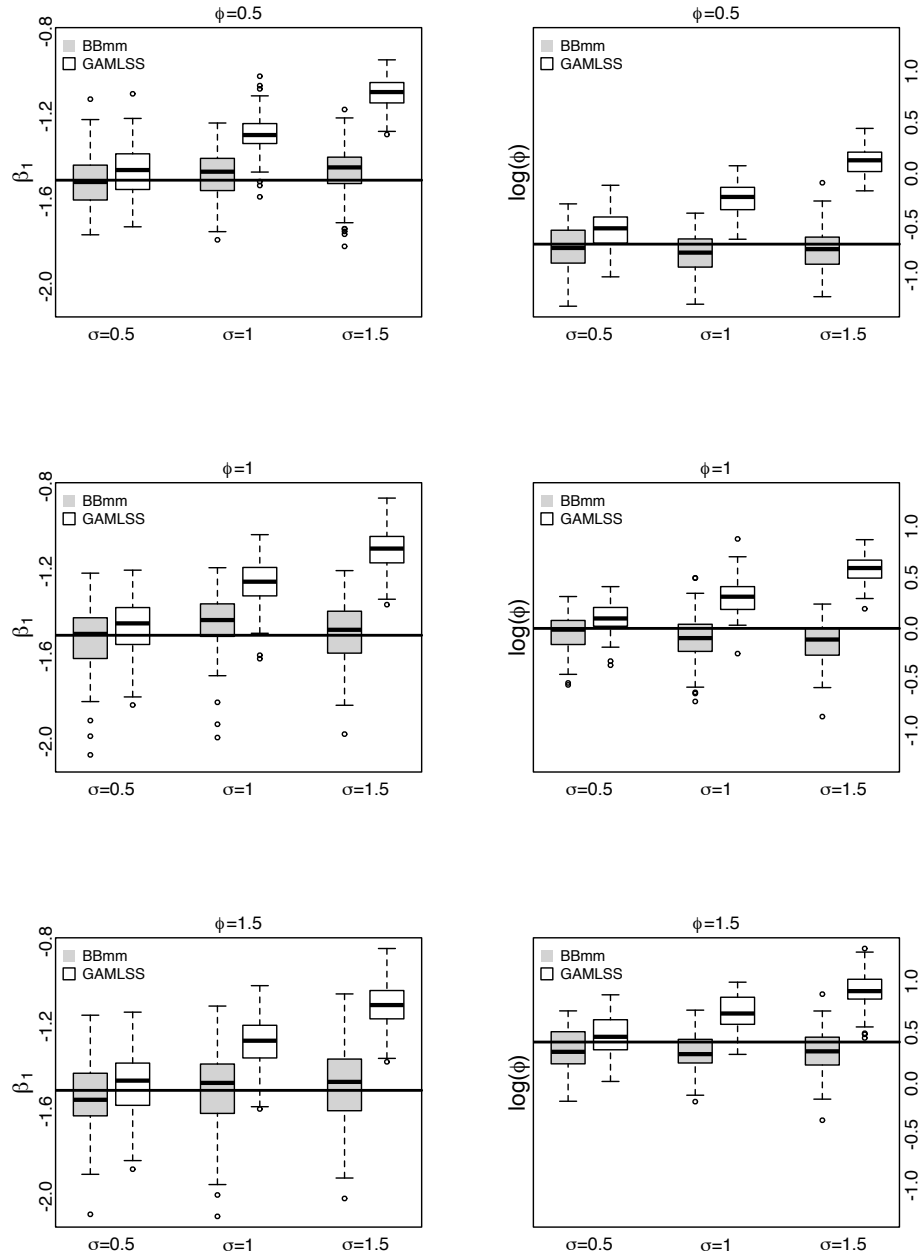


Figure 1: Box-plots of the estimates of  $\beta_1$  and  $\log(\phi)$  parameters in the simulation study. In the left panel the plots display the estimates for  $\beta_1$ , while in the right panel displays the estimates for  $\log(\phi)$ . Each group of box-plots describes three scenarios corresponding to all 9 scenarios.

```
library(PROreg)
library(gamlss)
library(numDeriv)
library(rootSolve)
```

$n = 500$			$\beta_1$					$\log(\phi)$				conv (%)
$\sigma$	$\phi$	Method	Mean	ESD	ASD	MSE	$C_{95\%}$	Mean	ESD	ASD	MSE	
1.5	1.5	BBmm	-1.413	0.093	0.079	0.016	75	0.389	0.126	0.112	0.016	100
		GAMLSS	-1.101	0.073	0.150	0.165	5	0.702	0.098	0.329	0.098	93
$n = 1000$			$\beta_1$					$\log(\phi)$				conv (%)
$\sigma$	$\phi$	Method	Mean	ESD	ASD	MSE	$C_{95\%}$	Mean	ESD	ASD	MSE	
1.5	1.5	BBmm	-1.453	0.067	0.056	0.006	77	0.387	0.082	0.080	0.007	100
		GAMLSS	-1.237	0.046	0.127	0.071	28	0.551	0.070	0.281	0.026	91

BBmm: beta-binomial mixed-effects model; ESD: empirical standard deviation; ASD: average standard deviation; MSE: mean square errors;  $C_{95\%}$ : Coverage probability of 95%; conv: Percentage of convergence.

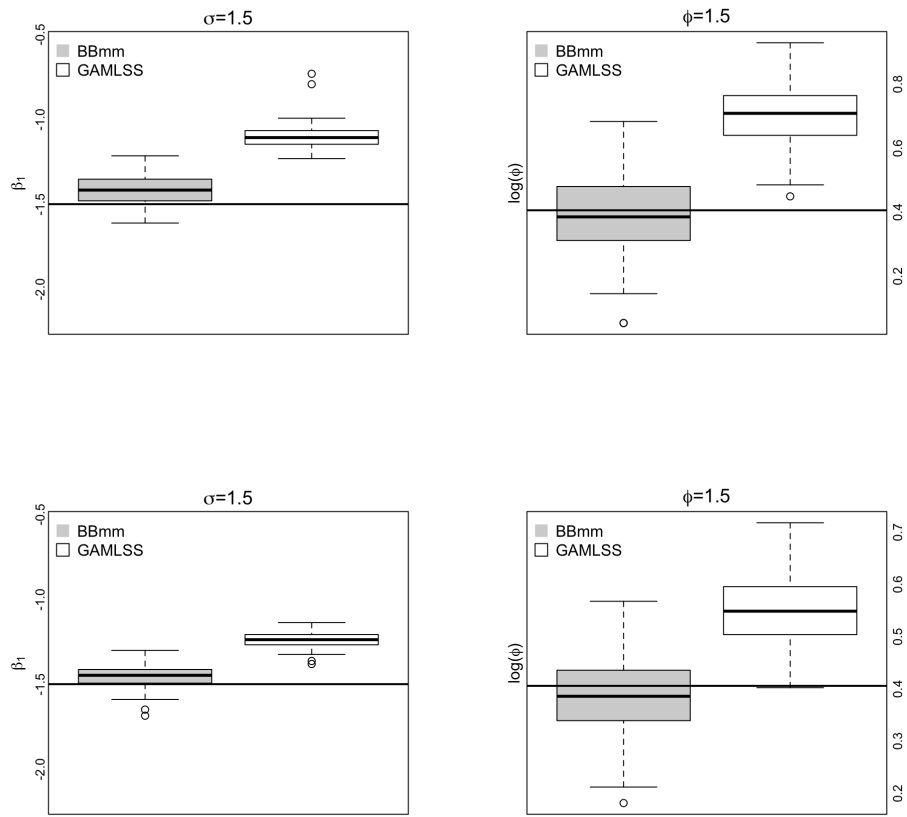


Figure 2: Box-plots of the estimates of  $\beta_1$  and  $\log(\phi)$  parameters with sample sizes of  $n = 500$  (top) and  $n = 1000$  (bottom). In the left panel the plots display the estimates for  $\beta_1$ , while in the right panel displays the estimates for  $\log(\phi)$ .

This is a function to fit BBmm and `gamlss` functions and save results

```
source("Simulation_function.R")
```

Set simulation parameters and generate

```

set.seed(8) # Set the seed
nSim <- 100 # No of simulations
nObs <- 200 # No of observations of the dependent variable for each simulation
nTrial <- sample(1:30,nObs,replace=T) # Random No of trials (form 1 to 30)
                                     # for each observation of the dependent variable
nRand <- 50 # No of random effects

x <- rnorm(nObs,1,2) # Simulate the covariate
z <- as.factor(PROreg::rBB(nObs, nRand-1,0.5,0.5)) # Simulate the levels of the random effects
nlevels(z)==nRand # Verify that the number of levels is equal to the number of random effects
X <- cbind(1,x) # Construct the model matrix for the fixed effects
Z <- model.matrix(~z-1) # Construct the model matrix for the random effects
beta <- c(1,-1.5)
beta.real <- beta

sigma <- 1.5 # Set the standard deviation of the random effects (=0.5,1,1.5)
phi <- 1.5 # Set the dispersion parameter of the beta-binomial distribution (=0.5,5,1.5)

u <- rnorm(nRand,0,sigma) # Simulate the random effects
dat <- data.frame(cbind(x,z)) # Generate a data frame with the covariate and random effects
dat$z <- as.factor(dat$z)

```

Simulate data and fit

```
out <- simulation.function(nSim,nObs,nTrial,beta,u,phi,sigma,dat)
```

Obtain results as in Table 2.

```

# Slope parameter

beta1 <- out$beta1.est # Estimates
mean.beta1 <- apply(beta1,2,mean) # Mean of the estimates
B.beta1 <- mean.beta1-beta.real[2] # Difference between the real value and mean estimate
V.beta1 <- apply(beta1,2,var) # Variance
EMS.beta1 <- B.beta1^2+V.beta1 # EMS
ESD.beta1 <- sqrt(V.beta1) # Empirical standard deviation
ASE.beta2 <- cbind(out$ASE.BBmm[2],out$ASE.GAMLSS[2]) # Average estandard deviation

out$prop.in.ic.beta # Coverage probability

# Logarithm of the dispersion parameter of the beta-binomial distribution (psi)

psi.real <- log(phi)
psi <- out$psi.est # Estimates
mean.psi <- apply(psi,2,mean) # Mean of the estimates
B.psi <- mean.psi-psi.real # Difference between real value and mean estimate
V.psi <- apply(psi,2,var) # Variance
EMS.psi <- B.psi^2+V.psi # EMS
ESD.psi <- sqrt(V.psi) # Empirical standard deviation
ASE.psi <- apply(out$psi.se,2,mean) # Average standard deviation

#Convergence
out$no.conv

```

## 4 R code for COPD data analysis

```
library(PROreg)
library(numDeriv)
library(rootSolve)
library(car)
library(Matrix)

load("COPD.rda")

nObs <- dim(COPD)[1]

Z.i <- model.matrix(~id-1,data=COPD)
Z.s <- model.matrix(~id:dy-1,data=COPD)
Z.uni <- cbind(Z.i,Z.s)

Model.RP <- BBmm1(fixed.formula=RP.~dy*cluster,Z=Z.uni,
                  nRandComp = c(543,543),m=4,data=dat,nDim=1,show = TRUE,maxiter=200)
Model.GH <- BBmm1(fixed.formula=GH.~dy*cluster,Z=Z.uni,
                  nRandComp = c(543,543),m=20,data=dat,nDim=1,show = TRUE,maxiter=200)
Model.RE <- BBmm1(fixed.formula=RE.~dy*cluster,Z=Z.uni,
                  nRandComp = c(543,543),m=3,data=dat,nDim=1,show = TRUE,maxiter=200)

library(gamlss)

GAMLSS.RP <- try(gamlss(cbind(RP.,4-RP.)~re(fixed=~dy*cluster,random=~1+dy|id),
                        family=BB(),data=dat))

smo.GAMLSS.RP <- getSmo(GAMLSS.RP)
summary(smo.GAMLSS.RP)
coef.gamlss.RP. <- smo.GAMLSS.RP$coefficients$fixed
coef.gamlss.RP.[1] <- coef.gamlss.RP.[1]+GAMLSS.RP$mu.coefficients[1]

psi.GAMLSS.RP <- GAMLSS.RP$sigma.coefficients
psi.gamlss.RP.se <- sqrt(summary(GAMLSS.RP)[4])

sigma.RP <- as.numeric(VarCorr(smo.GAMLSS.RP)[c(1,2),2])

GAMLSS.GH <- try(gamlss(cbind(GH.,20-GH.)~re(fixed=~dy*cluster,random=~dy|id),
                        family=BB(),data=dat))

smo.GAMLSS.GH <- getSmo(GAMLSS.GH)
summary(smo.GAMLSS.GH)
coef.gamlss.GH. <- smo.GAMLSS.GH$coefficients$fixed
coef.gamlss.GH.[1] <- coef.gamlss.GH.[1]+GAMLSS.GH$mu.coefficients[1]

psi.GAMLSS.GH <- GAMLSS.GH$sigma.coefficients
psi.gamlss.GH.se <- summary(GAMLSS.GH)[4]

sigma.GH <- as.numeric(VarCorr(smo.GAMLSS.GH)[c(1,2),2])

GAMLSS.RE <- try(gamlss(cbind(RE.,3-RE.)~re(fixed=~dy*cluster,random=~dy|id),
```

```

                                family=BB(),data=dat))

smo.GAMLSS.RE <- getSmo(GAMLSS.RE)
summary(smo.GAMLSS.RE)
coef.gamlss.RE. <- smo.GAMLSS.RE$coefficients$fixed
coef.gamlss.RE.[1] <- coef.gamlss.RE.[1]+GAMLSS.RE$mu.coefficients[1]

psi.GAMLSS.RE <- GAMLSS.RE$sigma.coefficients
psi.gamlss.RE.se <- sqrt(summary(GAMLSS.RE)[4])

sigma.RE <- as.numeric(VarCorr(smo.GAMLSS.RE)[c(1,2),2])

```