

Homework 4

Bayesian Statistical Methods

Michael Cotner

Department of Biomedical Engineering
The University of Texas at Austin
Email: mcotner@utexas.edu

Akshay Umashankar

Department of Educational Psychology
The University of Texas at Austin
Email: akshayumashankar38@utexas.edu

March 23, 2023

Introduction

The data in ‘mosquitofish.txt’ represent the number of surviving mosquitofish y_i (for $i = 1, \dots, n$) from $n = 32$ groups out of N_i initial fish in each group. For this homework assignment, we seek to construct a Binomial regression model to learn about covariates that may predict survival probability (θ_i). These covariates are standardized final mean length of surviving fish in group (L1), standardized duration in days of study (d), and also the product of the standardized L1 and d.

1 Using JAGS, develop the R code to fit the following Bayesian binomial regression model:

The model is as follows:

$$y_i \sim \text{Binom}(N_i, \theta_i)$$

$$\text{probit}(\theta_i) = \mathbf{x}_i' \boldsymbol{\beta}$$

$$\boldsymbol{\beta} \sim N(\boldsymbol{\mu}_\beta, \boldsymbol{\Sigma}_\beta)$$

The full JAGS implementation can be found in the code block in the appendix, but the relevant JAGS snippet is:

```
1 mosquito_jags <-"
2   model{
3     for(i in 1:n){
4       y[i] ~ dbin (theta[i] , N[i])
5       probit(theta[i]) = b0 + b1*X[i,2] + b2*X[i,3] + b3*(X[i
6     ],2]*X[i,3])
7     }
8     b0 ~ dnorm(mu, tau)
9     b1 ~ dnorm(mu, tau)
10    b2 ~ dnorm(mu, tau)
11    b3 ~ dnorm(mu, tau)
12  }
```

- 2 Use JAGS to fit the binomial regression model above to the mosquitofish data set using $y = N1$ as the response variable, $N = N0$, and the following combinations of the three covariates: (a) intercept, standardized $L1$, standardized d , and a product of the two covariates (as an interaction term) (b) intercept, standardized $L1$, standardized d (c) intercept, standardized $L1$ (d) intercept, standardized d (e) intercept only

Each implementation of the JAGS model ran successfully with good mixing and convergence to posteriors. The trace plots for each condition are shown in Figure 1.

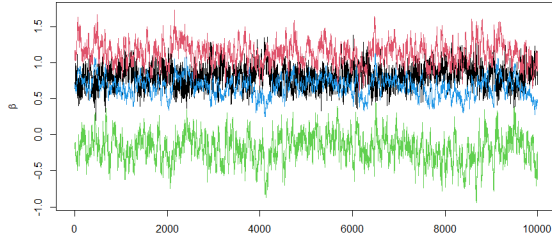
A table of the posterior parameters and their DIC values for each part are shown in Table 1

Model	DIC	β_0	β_1	β_2	β_3
a	77.448	1.351	-0.420	-2.423	2.687
b	95.881	1.387	0.847	-0.983	
c	134.916	1.310	0.095		
d	134.944	1.049		-0.085	
e	138.830	1.151			

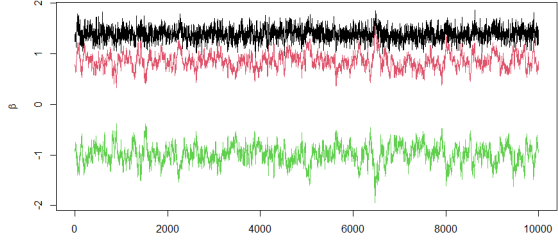
Table 1: Results of DIC and regression condition

- 3 Compare the DIC across model fits and make inference using the best scoring model according to DIC; include a violin plot for the regression coefficients.

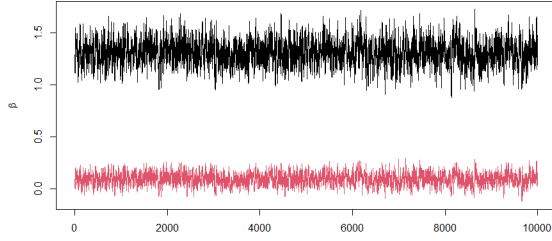
As shown in Table 1, the DIC values for c, d, and e are very close, suggesting that surviving fish L_1 in group or days d of study alone are not enough to predict y_i better



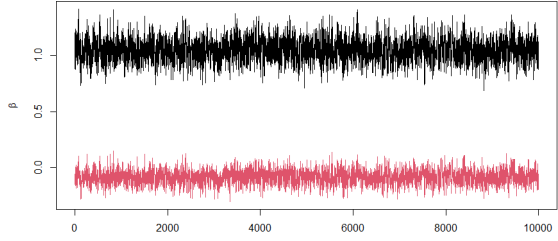
(a) β_0 : black, β_1 : red, β_2 : green, β_3 : blue,



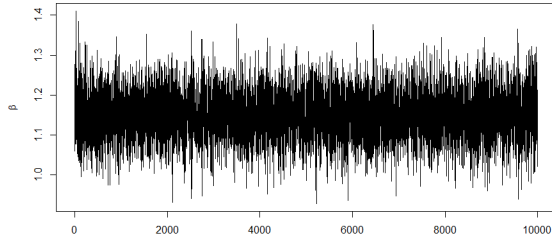
(b) β_0 : black, β_1 : red, β_2 : green



(c) β_0 : black, β_1 : red



(d) β_0 : black, β_2 : red



(e)

Figure 1: Trace plot

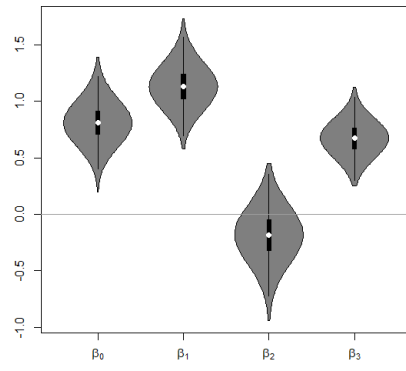
than a constant value (β_0) alone. This observation is further supported by the violin plots for c and d (Figure 2), where the slope values β_1 cross 0 and cannot be said to be statistically significant. However, together, surviving fish L_1 and days d together produce a better model, with some valid prediction of y_i . The values of β_1 and β_2 do not cross zero for part b, and the DIC for this implementation is much better. The predictive model further improves with the inclusion of an interaction term $L_1 * d$. The DIC is the lowest for this implementation, and the β_1 (L_1) and β_3 parameters ($L_1 * d$) show statistical significance to the predictive ability of the model.

References

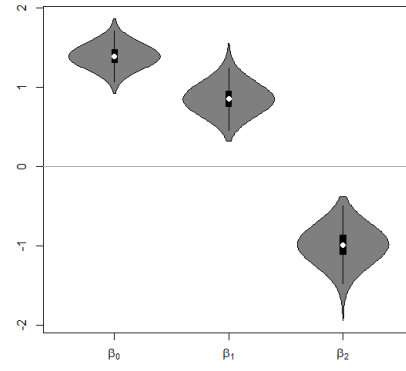
Soubeyand, S., R. Beaudouin, N. Desassis, and G. Monod. (2007). Model-based estimation of the link between the daily survival probability and a time-varying covariate, application to mosquitofish survival data. *Mathematical Biosciences*, 210: 508-522.

Author Contributions

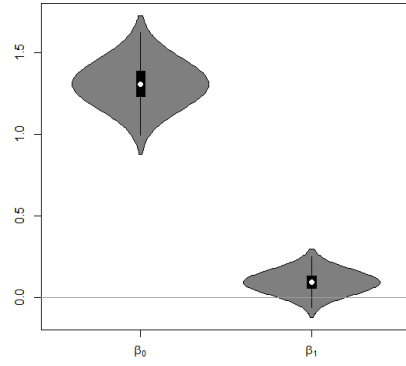
- AU and MC individually wrote a model implementation in R and analyzed the data set. Both authors compared answers and contributed to the homework solution document.



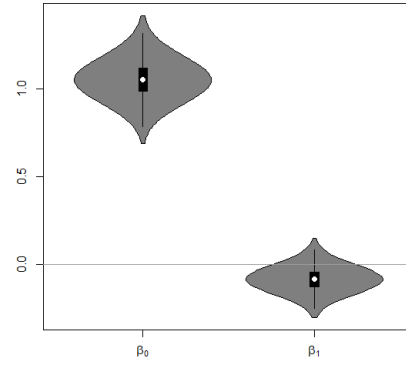
(a)



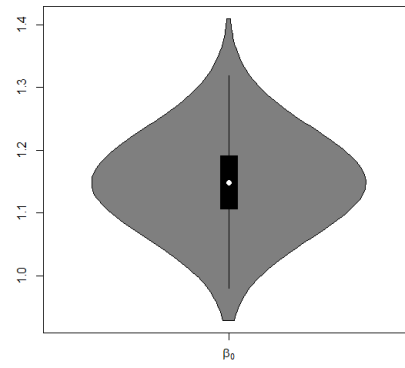
(b)



(c)



(d)



(e)

Figure 2: Violin plots for MCMC values for β parameters

APPENDIX A: AU's R Script

```
1 ### Homework 4
2
3 ### Libraries
4 library(rjags)
5 library(vioplot)
6 ###
7 ### Read the Data
8 ###
9
10
11
12 setwd("~/UT Austin/Spring 2023/Bayesian Stats/hw4/hw4")
13 mosquito=read.table("mosquitofish.txt",header=TRUE) # read in
    mosquitofish data
14 head(mosquito)
15
16
17 n = 32 # those groups of fish that were only counted twice
18 y = mosquito$N1[1:n]
19 N = mosquito$N0[1:n]
20 l1 = mosquito$L1[1:n]
21 d = mosquito$d[1:n]
22
23 l1_stand = scale(l1)[,1] #standardized L1 - mean length of surviving
    fish
24 d_stand = scale(d)[,1] #standardized Days of study
25
26 l1_d_product = l1_stand * d_stand
27
28
29 ## Jags Regression
30 # (a)
31
32 # Covariates - Intercept, stand l1, stand d, product of l1 and d
33 p = 4
34 X = matrix(1,n,p)
35 X[,2] = l1_stand
36 X[,3] = d_stand
37 X[,4] = l1_d_product
38
39 mosquito_jags <-"
40 model{
41   for(i in 1:n){
42     y[i] ~ dbin(theta[i],N[i])
43     probit(theta[i]) = b0 + b1*X[i,2] + b2*X[i,3] + b3*(X[i,2]*X[i
    ,3])
44   }
45   b0 ~ dnorm(mu,tau)
46   b1 ~ dnorm(mu,tau)
```

```

47     b2 ~ dnorm(mu,tau)
48     b3 ~ dnorm(mu,tau)
49 }
50 "
51 mod <- textConnection(mosquito_jags)
52 #
53 # mu.beta=rep(0,p)
54 # Sig.beta=100*diag(p)
55 # Tau.beta=solve(Sig.beta)
56
57 mu=0
58 tau=1/2.25 #Precision. 1 / sigma^2
59 m.out<-jags.model(mod,data=list('y'=y,'n'=n, 'N'=N, 'X'=X,'mu'=mu,'
    tau'=tau),n.chains=1,n.adapt=0) # build model and algorithm
60
61 n.mcmc=10000
62 n.burn=round(.2*n.mcmc)
63 update(m.out,n.burn) # perform burn-in
64 m.samples=jags.samples(m.out,c('b0','b1','b2','b3'),n.mcmc) # fit
    model post-burn-in
65
66 #Trace Plot for (a)
67 beta.post.mat=cbind(m.samples$b0[1,,1],m.samples$b1[1,,1],m.samples$
    b2[1,,1],m.samples$b3[1,,1])
68 matplot(beta.post.mat,type="l",lty=1,ylab=bquote(beta))
69
70 #Violin plot for (a)
71 vioplot(data.frame(beta.post.mat),names=expression(beta[0],beta[1],
    beta[2],beta[3]))
72 abline(h=0,col=8)
73
74 apply(beta.post.mat,2,mean) # marginal posterior means for beta
75 apply(beta.post.mat,2,sd) # marginal posterior sd for beta
76 apply(beta.post.mat,2,quantile,c(0.025,.975)) # marginal posterior
    95% CI for beta
77
78
79 # Covariates - Intercept, stand l1, stand d
80 # (b)
81
82 mosquito_jags_b <- "
83 model{
84     for(i in 1:n){
85         y[i] ~ dbin(theta[i],N[i])
86         probit(theta[i]) = b0 + b1*X[i,2] + b2*X[i,3]
87     }
88     b0 ~ dnorm(mu,tau)
89     b1 ~ dnorm(mu,tau)
90     b2 ~ dnorm(mu,tau)
91 }
92 "

```



```

93 mod_b <- textConnection(mosquito_jags_b)
94 #
95 # mu.beta=rep(0,p)
96 # Sig.beta=100*diag(p)
97 # Tau.beta=solve(Sig.beta)
98
99 mu=0
100 tau=1/2.25 #Precision. 1 / sigma^2
101 m.out_b<-jags.model(mod_b,data=list('y'=y,'n'=n, 'N'=N, 'X'=X,'mu'=
      mu,'tau'=tau),n.chains=1,n.adapt=0) # build model and algorithm
102
103 n.mcmc=10000
104 n.burn=round(.2*n.mcmc)
105 update(m.out_b,n.burn) # perform burn-in
106 m.samples_b=jags.samples(m.out_b,c('b0','b1','b2'),n.mcmc) # fit
      model post-burn-in
107
108 #Trace Plot for (b)
109 beta.post.mat_b=cbind(m.samples_b$b0[1,,1],m.samples_b$b1[1,,1],m.
      samples_b$b2[1,,1])
110 matplot(beta.post.mat_b,type="l",lty=1,ylab=bquote(beta))
111
112 #Violin plot for (b)
113 vioplot(data.frame(beta.post.mat_b),names=expression(beta[0],beta
      [1],beta[2]))
114 abline(h=0,col=8)
115
116 apply(beta.post.mat_b,2,mean) # marginal posterior means for beta
117 apply(beta.post.mat_b,2,sd) # marginal posterior sd for beta
118 apply(beta.post.mat_b,2,quantile,c(0.025,.975)) # marginal posterior
      95% CI for beta
119
120 # Covariates - Intercept, stand l1
121 # (c)
122
123 mosquito_jags_c <- "
124   model{
125     for(i in 1:n){
126       y[i] ~ dbin(theta[i],N[i])
127       probit(theta[i]) = b0 + b1*X[i,2]
128     }
129     b0 ~ dnorm(mu,tau)
130     b1 ~ dnorm(mu,tau)
131   }
132 "
133 mod_c <- textConnection(mosquito_jags_c)
134 #
135 # mu.beta=rep(0,p)
136 # Sig.beta=100*diag(p)
137 # Tau.beta=solve(Sig.beta)
138

```

```

139 mu=0
140 tau=1/2.25 #Precision. 1 / sigma^2
141 m.out_c<-jags.model(mod_c,data=list('y'=y, 'n'=n, 'N'=N, 'X'=X, 'mu'=
      mu, 'tau'=tau),n.chains=1,n.adapt=0) # build model and algorithm
142
143 n.mcmc=10000
144 n.burn=round(.2*n.mcmc)
145 update(m.out_c,n.burn) # perform burn-in
146 m.samples_c=jags.samples(m.out_c,c('b0','b1'),n.mcmc) # fit model
      post-burn-in
147
148 #Trace Plot for (c)
149 beta.post.mat_c=cbind(m.samples_c$b0[1,,1],m.samples_c$b1[1,,1],m.
      samples_c$b2[1,,1])
150 matplot(beta.post.mat_c,type="l",lty=1,ylab=bquote(beta))
151
152 #Violin plot for (c)
153 vioplot(data.frame(beta.post.mat_c),names=expression(beta[0],beta
      [1]))
154 abline(h=0,col=8)
155
156 apply(beta.post.mat_c,2,mean) # marginal posterior means for beta
157 apply(beta.post.mat_c,2,sd) # marginal posterior sd for beta
158 apply(beta.post.mat_c,2,quantile,c(0.025,.975)) # marginal posterior
      95% CI for beta
159
160
161 # Covariates - Intercept, stand d
162 # (d)
163
164 mosquito_jags_d <-"
165   model{
166     for(i in 1:n){
167       y[i] ~ dbin(theta[i],N[i])
168       probit(theta[i]) = b0 + b1*X[i,3]
169     }
170     b0 ~ dnorm(mu,tau)
171     b1 ~ dnorm(mu,tau)
172   }
173 "
174 mod_d <- textConnection(mosquito_jags_d)
175 #
176 # mu.beta=rep(0,p)
177 # Sig.beta=100*diag(p)
178 # Tau.beta=solve(Sig.beta)
179
180 mu=0
181 tau=1/2.25 #Precision. 1 / sigma^2
182 m.out_d<-jags.model(mod_d,data=list('y'=y, 'n'=n, 'N'=N, 'X'=X, 'mu'=
      mu, 'tau'=tau),n.chains=1,n.adapt=0) # build model and algorithm
183

```

```

184 n.mcmc=10000
185 n.burn=round(.2*n.mcmc)
186 update(m.out_d,n.burn) # perform burn-in
187 m.samples_d=jags.samples(m.out_d,c('b0','b1'),n.mcmc) # fit model
    post-burn-in
188
189 #Trace Plot for (d)
190 beta.post.mat_d=cbind(m.samples_d$b0[1,,1],m.samples_d$b1[1,,1])
191 matplot(beta.post.mat_d,type="l",lty=1,ylab=bquote(beta))
192
193 #Violin plot for (d)
194 vioplot(data.frame(beta.post.mat_d),names=expression(beta[0],beta
    [1]))
195 abline(h=0,col=8)
196
197 apply(beta.post.mat_d,2,mean) # marginal posterior means for beta
198 apply(beta.post.mat_d,2,sd) # marginal posterior sd for beta
199 apply(beta.post.mat_d,2,quantile,c(0.025,.975)) # marginal posterior
    95% CI for beta
200
201 # Covariates - Intercept ONLY
202 # (e)
203
204 mosquito_jags_e <-"
205   model{
206     for(i in 1:n){
207       y[i] ~ dbin(theta[i],N[i])
208       probit(theta[i]) = b0
209     }
210     b0 ~ dnorm(mu,tau)
211   }
212 "
213 mod_e <- textConnection(mosquito_jags_e)
214 #
215 # mu.beta=rep(0,p)
216 # Sig.beta=100*diag(p)
217 # Tau.beta=solve(Sig.beta)
218
219 mu=0
220 tau=1/2.25 #Precision. 1 / sigma^2
221 m.out_e<-jags.model(mod_e,data=list('y'=y,'n'=n, 'N'=N, 'X'=X,'mu'=
    mu,'tau'=tau),n.chains=1,n.adapt=0) # build model and algorithm
222
223 n.mcmc=10000
224 n.burn=round(.2*n.mcmc)
225 update(m.out_e,n.burn) # perform burn-in
226 m.samples_e=jags.samples(m.out_e,c('b0'),n.mcmc) # fit model post-
    burn-in
227
228 #Trace Plot for (e)
229 beta.post.mat_e=cbind(m.samples_e$b0[1,,1])

```

```

230 matplot(beta.post.mat_e,type="l",lty=1,ylab=bquote(beta))
231
232 #Violin plot for (e)
233 vioplot(data.frame(beta.post.mat_e),names=expression(beta[0]))
234 abline(h=0,col=8)
235
236 apply(beta.post.mat_e,2,mean) # marginal posterior means for beta
237 apply(beta.post.mat_e,2,sd) # marginal posterior sd for beta
238 apply(beta.post.mat_e,2,quantile,c(0.025,.975)) # marginal posterior
    95% CI for beta
239
240 #
    -----

241 ### PLOTS and Quantiles
242
243 ## All Trace Plots
244 matplot(beta.post.mat,type="l",lty=1,ylab=bquote(beta)) # (a)
245 matplot(beta.post.mat_b,type="l",lty=1,ylab=bquote(beta)) # (b)
246 matplot(beta.post.mat_c,type="l",lty=1,ylab=bquote(beta)) # (c)
247 matplot(beta.post.mat_d,type="l",lty=1,ylab=bquote(beta)) # (d)
248 matplot(beta.post.mat_e,type="l",lty=1,ylab=bquote(beta)) # (e)
249
250 ## All Violin Plots
251
252 vioplot(data.frame(beta.post.mat),names=expression(beta[0],beta[1],
    beta[2],beta[3])) # (a)
253 abline(h=0,col=8)
254
255 vioplot(data.frame(beta.post.mat_b),names=expression(beta[0],beta
    [1],beta[2])) # (b)
256 abline(h=0,col=8)
257
258 vioplot(data.frame(beta.post.mat_c),names=expression(beta[0],beta
    [1])) # (c)
259 abline(h=0,col=8)
260
261 vioplot(data.frame(beta.post.mat_d),names=expression(beta[0],beta
    [1])) # (d)
262 abline(h=0,col=8)
263
264 vioplot(data.frame(beta.post.mat_e),names=expression(beta[0]))
    # (e)
265 abline(h=0,col=8)
266
267 ## All Quantiles
268 apply(beta.post.mat,2,quantile,c(0.025,.975)) # (a)
269 apply(beta.post.mat_b,2,quantile,c(0.025,.975)) # (b)
270 apply(beta.post.mat_c,2,quantile,c(0.025,.975)) # (c)
271 apply(beta.post.mat_d,2,quantile,c(0.025,.975)) # (d)
272 apply(beta.post.mat_e,2,quantile,c(0.025,.975)) # (e)

```

```

273
274
275 ## DIC Calculations
276
277 # (a) DIC = 74.6349860620726
278 # (b) DIC = 97.1322909927127
279 # (c) DIC = 136.613174557372
280 # (d) DIC = 137.52021356181
281 # (e) DIC = 139.469354229737
282
283 #(a)
284
285 postbeta = apply(beta.post.mat, 2, mean)
286 posttheta = pnorm(X%%postbeta)
287 Dhat = mean(-2*sum(log(dbinom(y,N,posttheta))))
288 pD = 2 * sum(log(mean(exp(beta.post.mat), 2)))
289 DIC=Dhat+2*pD
290 print(paste("DIC =",DIC))
291 # "DIC = 74.6349860620726"
292
293 #(b)
294 postbeta = apply(beta.post.mat_b, 2, mean)
295 posttheta = pnorm(X[, -4]%%postbeta)
296 Dhat = mean(-2*sum(log(dbinom(y,N,posttheta))))
297 pD = 2 * sum(log(mean(exp(beta.post.mat), 2)))
298 DIC=Dhat+2*pD
299 print(paste("DIC =",DIC))
300 # "DIC = 97.1322909927127"
301
302 #(c)
303 postbeta = apply(beta.post.mat_c, 2, mean)
304 posttheta = pnorm(X[, 1:2]%%postbeta)
305 Dhat = mean(-2*sum(log(dbinom(y,N,posttheta))))
306 pD = 2 * sum(log(mean(exp(beta.post.mat), 2)))
307 DIC=Dhat+2*pD
308 print(paste("DIC =",DIC))
309 # "DIC = 136.613174557372"
310
311 #(d)
312 postbeta = apply(beta.post.mat_d, 2, mean)
313 posttheta = pnorm(X[, c(1,3)]%%postbeta)
314 Dhat = mean(-2*sum(log(dbinom(y,N,posttheta))))
315 pD = 2 * sum(log(mean(exp(beta.post.mat), 2)))
316 DIC=Dhat+2*pD
317 print(paste("DIC =",DIC))
318 # "DIC = 137.52021356181"
319
320 #(e)
321 postbeta = apply(beta.post.mat_e, 2, mean)
322 posttheta = pnorm(X[, 1]*postbeta)
323 Dhat = mean(-2*sum(log(dbinom(y,N,posttheta))))

```

```
324 pD = 2 * sum(log(mean(exp(beta.post.mat), 2)))
325 DIC=Dhat+2*pD
326 print(paste("DIC =",DIC))
327 # "DIC = 139.469354229737"
```

APPENDIX B: MC's R Script

```
1 library(rjags)
2 mosquitofish = read.table(file = 'RStudio/bayesian/mosquitofish.txt'
3   , header=TRUE)
4
5 ### a
6 X = data.matrix(mosquitofish[c('L1', 'd')])
7 X = cbind(X, X[,1]*X[,2])
8 xavg = c(0, apply(X, 2, mean))
9 xsd = c(1, apply(X, 2, sd))
10 X = scale(X)
11 X = cbind(1, X)
12
13 ### b
14 X = data.matrix(mosquitofish[c('L1', 'd')])
15 X = scale(X)
16 X = cbind(1, X)
17
18 ### c
19 X = data.matrix(mosquitofish[c('L1')])
20 X = scale(X)
21 X = cbind(1, X)
22
23 ### d
24 X = data.matrix(mosquitofish[c('d')])
25 X = scale(X)
26 X = cbind(1, X)
27
28 ### e
29 X = matrix(1, n)
30
31 m.jags <-"
32   model{
33     for(i in 1:n)
34     {
35       y[i,] ~ dbin(theta[i], N[i,])
36       probit(theta[i]) = inprod(X[i,], b)
37     }
38     b ~ dmnorm(mu_b[], tau_b[,])
39   }
40 "
41
42 mod<-textConnection(m.jags)
43
44 y = data.matrix(mosquitofish['N1'])
45 N = data.matrix(mosquitofish['NO'])
46 n = dim(y)[1]
47
48 nx = dim(X)[2]
```

```

49 mu_b = apply(X, 2, mean)
50 sig_b = diag(nx)*((apply(X, 2, sd)^2)+1)
51 tau_b = solve(sig_b)
52 m.out<-jags.model(mod,data=list('y'=y, 'X'=X, 'n'=n, 'N'=N, 'mu_b'=mu_b,
    'tau_b'=tau_b),n.chains=1,n.adapt=0)
53 n.mcmc=100000
54 n.burn=round(.1*n.mcmc)
55 update(m.out,n.burn) # perform burn-in
56 m.samples=jags.samples(m.out,c('theta','b'),n.mcmc)
57
58 par(mfrow=c(nx,1))
59 for(i in 1:nx)
60 {
61   plot(m.samples$b[i,,],type="l", ylab=paste("b_",i,sep=""))
62 }
63 #plot(theta,type="l", ylab="tau")
64
65 postbeta = apply(m.samples$b, 1, mean)
66 posttheta = pnorm(X%%postbeta)
67 Dhat = mean(-2*sum(log(dbinom(y,N,posttheta))))
68 pD = 2 * sum(log(mean(exp(m.samples$b), 2)))
69 DIC=Dhat+2*pD
70 print(paste("DIC =",DIC))
71
72
73 #post_beta_unscale = postbeta*xsd
74
75 library(vioplot)
76 vioplot(t(m.samples$b[,,]), names = list("b_0", "b_1", "b_2", "b_3")
    )
77 abline(h=0,col=8)

```