# Applied Bayesian Analysis : NCSU ST 540

Homework 7

*Bruce Campbell*

---

In this assignment we will performing random slopes logistic regression in JAGS using the Gambia data described in `http://www4.stat.ncsu.edu/~reich/ABA/code/GLM` Let $Y_i$ be the binary response for individual $i$ , and let $\nu_i \in 1 \cdots 65$ denote the village of individual $i$ Let $X_i = 1$ if individual $i$ regularly sleeps under a bed-net and $X_i = 0$ otherwise. Fit the model

$$logit(P(Y_i = 1)) = \alpha_{\nu_i} + X_i \beta_{\nu_i}$$

where $\alpha_{\nu_i}$ and $\beta_{\nu_i}$ are the intercept and slope for village $j$ The priors (independent over village and with each other) are

$$\alpha_{\nu_i} \sim Normal(\mu_a, \sigma_a^2)$$

and

$$\beta_{\nu_i} \sim Normal(\mu_b, \sigma_b^2)$$

Choose uninformative priors for $\mu_a, \sigma_a^2, \mu_b, \sigma_b^2$

In your report address the follow questions: (1) Scientifically, why might the effect of bed-net vary by village? (2) Did the MCMC algorithm converge? (3) Do you see evidence that the slopes and/or intercepts vary by village? (4) Which village has the largest intercept? Slope? Does this agree with the data in these villages?

```
library(rjags)
library(coda)
library(modeest)

DEBUG <- FALSE
if(DEBUG)
{
nSamples <- 10000
n.chains <- 1
} else
{
nSamples <- 10000
n.chains <- 1
}

load("gambia.RData")

X.net <- as.numeric((X$netuse==1) | (X$treated==1))
Y <- pos
n <- length(X.net)
```

```r
df <- data.frame(cbind(X.net,village,pos))
names(df) <- c("net","village","pos")
numVillages <- length(unique(df$village))
villages <- df$village

#boxplot(pos ~ village, data = df)
#plot(df$village,df$pos)
#ggplot(df, aes(x=village, y=net, color=pos))
#plot(df$village)
#village.counts <- unlist(table(df$village))

model_string.logistic_random_slopes <- "model{
   # Likelihood
   for(i in 1:n){
      Y[i]    ~ dbern(q[i])
      logit(q[i]) <- beta[villages[i],1] + beta[villages[i],2]*X.net[i]
   }

   # Random effects
   for(j in 1:numVillages){
    beta[j,1] ~dnorm(0,0.1)
    beta[j,2] ~dnorm(0,0.1)
   }

    for(j in 1:numVillages){
      pred[j] <- beta[j,1] + beta[j,2]
    }
  }"


model.logistic_random_slopes <- jags.model(textConnection(model_string.logistic_random_slopes)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 2035
##    Unobserved stochastic nodes: 130
##    Total graph size: 6625
##
## Initializing model
```

```r
update(model.logistic_random_slopes, nSamples, progress.bar="none"); # Burnin
samp.coeff.logistic_random_slopes <- coda.samples(model.logistic_random_slopes, variable.names=


sum.logistic_random_slopes <-  summary(samp.coeff.logistic_random_slopes)
quantiles<-sum.logistic_random_slopes$quantiles
```

```r
left.05.quantile.sign  <- sign(quantiles[,1])==-1
right.95.quantile.sign <- sign(quantiles[,5])==1
significant <- xor(left.05.quantile.sign ,right.95.quantile.sign)
beta.significant <- quantiles[significant,]


pander(data.frame(beta.significant), caption = "significant ")
```

Table 1: significant

| | X2.5. | X25. | X50. | X75. | X97.5. |
|---|---|---|---|---|---|
| **beta**[**8,1**] | -6.549 | -4.069 | -3.044 | -2.162 | -0.7341 |
| **beta**[**9,1**] | -3.095 | -2.444 | -2.144 | -1.866 | -1.377 |
| **beta**[**10,1**] | -1.839 | -1.199 | -0.9 | -0.61 | -0.07173 |
| **beta**[**16,1**] | -4.113 | -2.519 | -1.814 | -1.179 | -0.1003 |
| **beta**[**40,1**] | -4.369 | -2.9 | -2.257 | -1.693 | -0.7715 |
| **beta**[**44,1**] | -6.083 | -3.648 | -2.649 | -1.771 | -0.4612 |
| **beta**[**51,1**] | 0.07391 | 0.6129 | 0.9084 | 1.209 | 1.833 |
| **beta**[**56,1**] | 0.2184 | 1.549 | 2.425 | 3.388 | 5.668 |
| **beta**[**60,1**] | 0.5581 | 1.039 | 1.302 | 1.573 | 2.151 |
| **beta**[**61,1**] | 0.1112 | 0.6795 | 0.9843 | 1.307 | 1.985 |
| **beta**[**64,1**] | 0.6911 | 1.76 | 2.405 | 3.152 | 4.945 |
| **beta**[**45,2**] | -3.914 | -2.662 | -2.103 | -1.605 | -0.7383 |
| **beta**[**51,2**] | -4.8 | -3.021 | -2.221 | -1.479 | -0.1849 |

```r
credible.widths <- beta.significant[,5]-beta.significant[,1]

pander(data.frame(credible.widths), caption = "credible widths  ")
```

Table 2: credible widths

| | credible.widths |
|---|---|
| **beta**[**8,1**] | 5.815 |
| **beta**[**9,1**] | 1.718 |
| **beta**[**10,1**] | 1.767 |
| **beta**[**16,1**] | 4.013 |
| **beta**[**40,1**] | 3.597 |
| **beta**[**44,1**] | 5.621 |
| **beta**[**51,1**] | 1.759 |
| **beta**[**56,1**] | 5.449 |
| **beta**[**60,1**] | 1.593 |
| **beta**[**61,1**] | 1.874 |
| **beta**[**64,1**] | 4.254 |
| **beta**[**45,2**] | 3.176 |
| **beta**[**51,2**] | 4.615 |

```
if (DEBUG)
  {
  autocorr.plot(samp.coeff.logistic_random_slopes)

  plot(samp.coeff.logistic_random_slopes)

  #Sample again and estimate posterior means and MAP posterior modes.
  samp.coeff.logistic_random_slopes.jags <- jags.samples(model.logistic_random_slopes, variabl
  posterior_means.logistic_random_slopes <- lapply(samp.coeff.logistic_random_slopes.jags, appl
  pander(posterior_means.logistic_random_slopes, caption = "posterior means second sample")

  posterior_modes.logistic_random_slopes <- lapply(samp.coeff.logistic_random_slopes.jags, appl
  posterior_modes.logistic_random_slopes

  if(n.chains>1)
  {
  gelman.plot(samp.coeff)
  }
}
```