Applied Bayesian Analysis : NCSU ST 540

Homework 5

Bruce Campbell

---

let Y_i be the number of concussions from team $i = 1, ..., 32$. The model is

$$Y_i|\lambda_i \sim Poisson(\lambda_i)$$

and the prior is

$$\lambda_i|\theta \sim Gamma(1, \theta)$$

where

$$\theta \sim Gamma(0.1, 0.1)$$

.

Derive the full conditional distribution of λ_1

Assuming independence of Y_i we can write the full joint distribution as

$$P(Y_1, ..., Y_n, \lambda_1, ...\lambda_n, \theta) \propto \prod_{i=1}^{n} P(Y_i|\lambda_i)P(\lambda_i|\theta)P(\theta)$$

Now

$$P(\lambda_1|Y_1, ..., Y_n, \lambda_2, ...\lambda_n, \theta) \propto P(Y_1|\lambda_1)P(\lambda_1|\theta)P(\theta) \quad \prod_{i=2}^{n} P(Y_i|\lambda_i)P(\lambda_i|\theta)$$

Putting the expressions for the densities in here and dropping the product terms on the right hand side unrelated to λ_1 we have that

$$P(\lambda_1|Y_1, ..., Y_n, \lambda_2, ...\lambda_n, \theta) \propto \frac{\lambda_1^{y_1}}{y_1!} e^{-\lambda_1} \quad \theta e^{-\theta\lambda_1} \quad \frac{(0.1)^{0.1}}{\Gamma(0.1)} \theta^{0.1-1} e^{-0.1\ \theta} \propto \lambda_1^{(y_1+1)-1} e^{-\lambda_1(1+\theta)}$$

The last experssion we recognise as the kernel of a $Gamma(y_1 + 1, 1 + \theta)$ distribution.

Derive the full conditional distribution of θ

From above, we have that

$$P(\theta|Y_1, ..., Y_n, \lambda_1, ...\lambda_n) \propto P(\theta) \quad \prod_{i=1}^{n} P(\lambda_i|\theta)$$

and putting the expression for the densities in we have

$$P(\theta|Y_1,...,Y_n,\lambda_1,...\lambda_n) \propto \prod_{i=1}^{n} \theta e^{-\theta\lambda_i} \quad \frac{(0.1)^{0.1}}{\Gamma(0.1)} \theta^{0.1-1} e^{-0.1\,\theta} \propto \quad \theta^{n-.9}\ e^{\theta(0.1+\sum\lambda_i)}$$

We recognise this expression as the kernel of a $Gamma(n+0.1, 0.1+\sum\lambda_i)$ density.

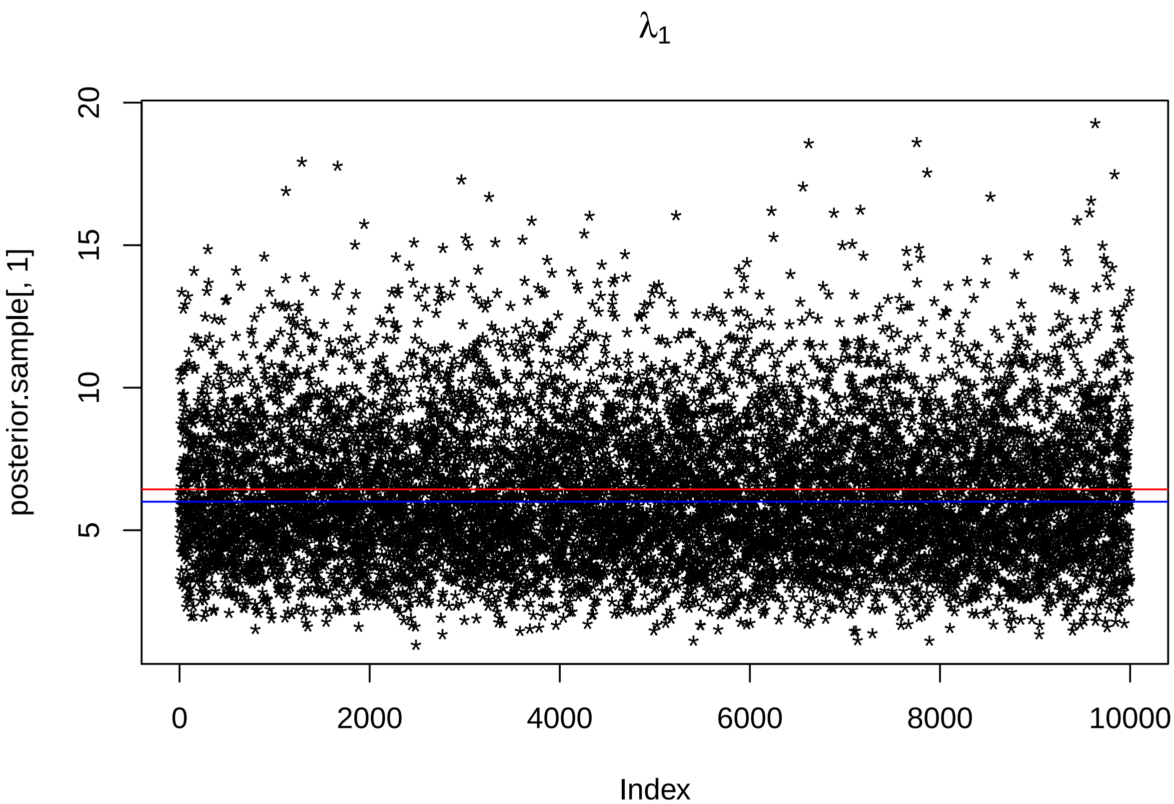
Write Gibbs sampling code to draw samples from the joint distribution of $(\lambda_1,...,\lambda_{32},\theta)$.

```
df <- read.csv("ConcussionsByTeamAndYear.csv",
    header = TRUE)
df$sum <- df$X2012 + df$X2013
n <- nrow(df)
M <- 10000  #Burn in
N <- 10000  #Number of draws

posterior.sample <- matrix(nrow = N, ncol = n +
    1)
# Set the initial values to the means
# of the respective distributions

theta.0 <- 1  #Mean of Gamma(.1,.1)

lambda.0 <- matrix(data = rep(1, n), n,
    1)  # Set to mean of Gamma(1,theta.0)


# Initialize outside loop
theta.t <- 1
lambda.t <- matrix(data = rep(1, n), n,
    1)

for (j in 1:M + N) {
    theta.t <- rgamma(1, shape = n + 0.1,
        rate = 0.1 + sum(lambda.t))
    for (i in 1:n) {
        y.i <- df$sum[i]
        lambda.t[i] <- rgamma(1, y.i +
            1, 1 + theta.t)
    }
    if (j > M) {
        sample <- c(lambda.t, theta.t)
        posterior.sample[j - M, ] <- sample
    }
}
```

Show trace plots of the samples for λ_1 and θ.

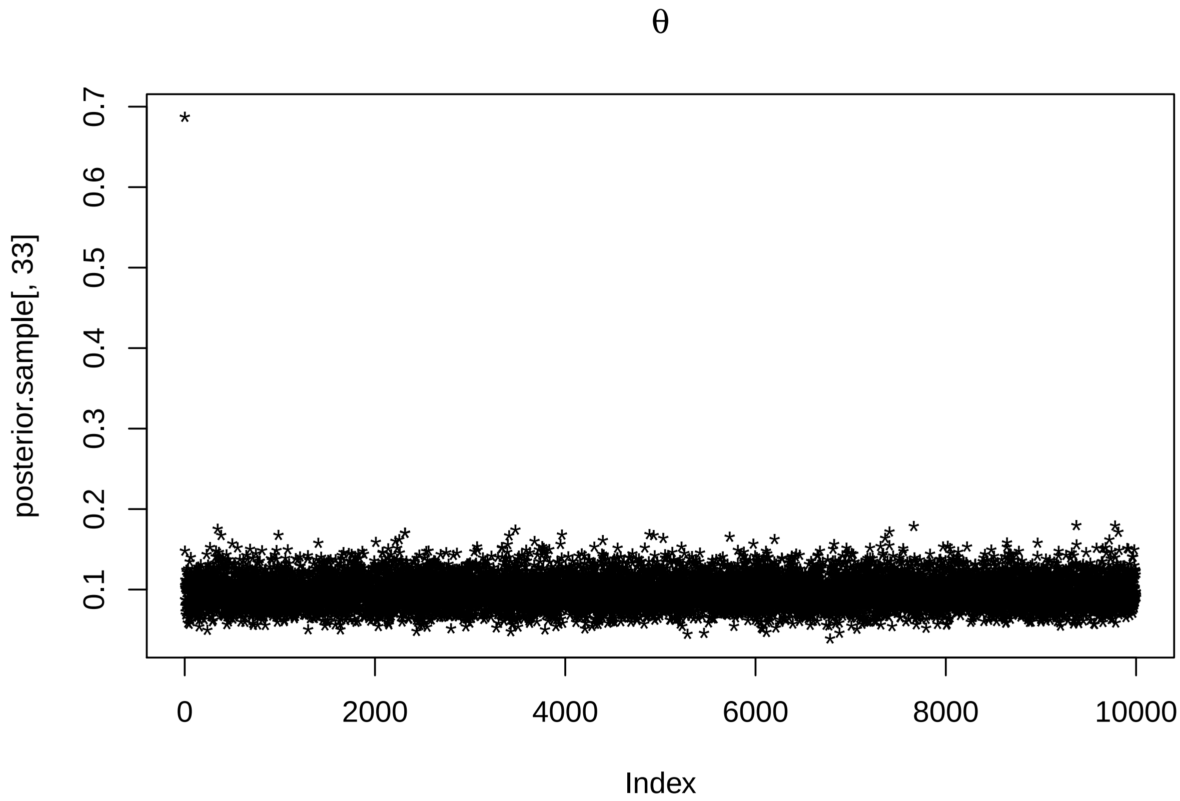
Trace plot for λ_1

```
plot(posterior.sample[, 1], main = TeX("$\\lambda_1$"),
    pch = "*")
abline(h = mean(posterior.sample[, 1]),
    col = "red")
abline(h = mean(df$sum[1]), col = "blue")
```



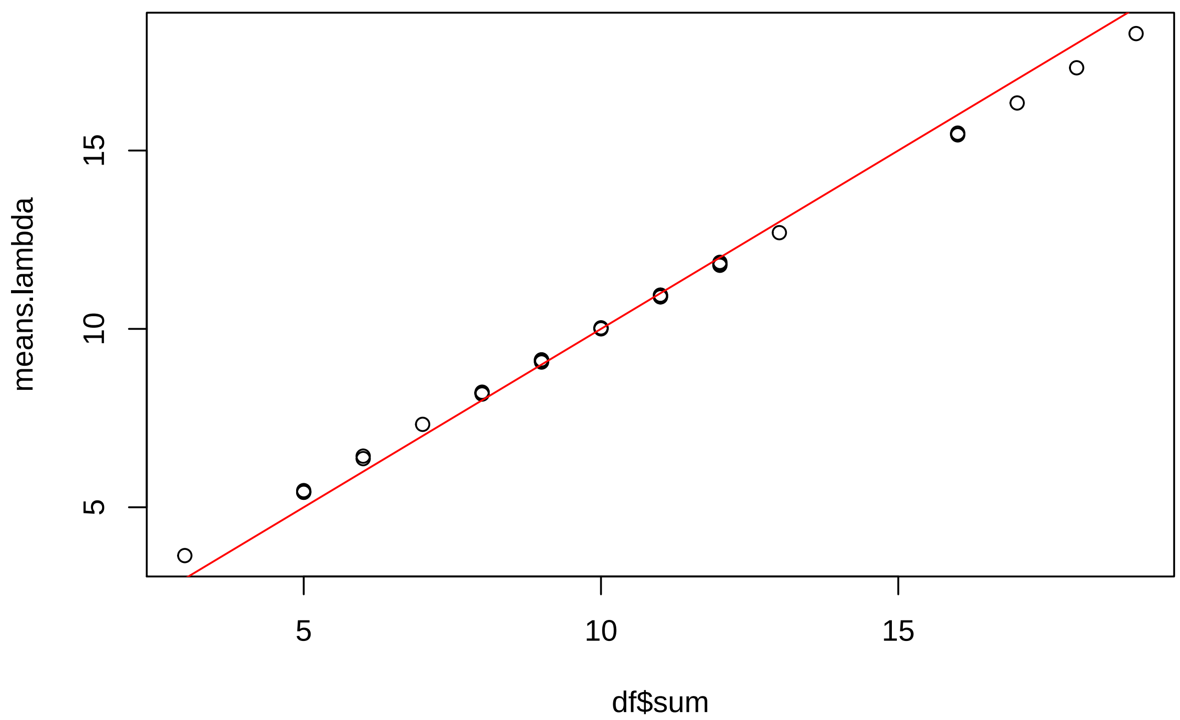
Trace plot for θ

```
plot(posterior.sample[, 33], main = TeX("$\\theta$"),
    pch = "*")
```



(5) Plot the estimated posterior mean of the λ_i versus Y_i and comment on whether the code is returning reasonable estimates. Turn in your solution on one piece of paper. Also, turn in MCMC code on a second piece of paper stapled to the solution.

```
means.lambda <- as.list(colMeans(posterior.sample))
means.lambda[[33]] <- NULL
plot(df$sum, means.lambda)
abline(a = 0, b = 1, col = "red")
```



We see good alignment between the values Y_i and λ_i - remember $Y \sim Pisson(\lambda) \implies E[Y] = \lambda$

Test - Let's validate with Jags

```
library(rjags)
library(coda)
a      <- .1
b      <- .1
model_string <- "model{

  for(i in 1:n)
  {
    lambda[i] ~ dgamma(1,theta)
    # Likelihood
    Y[i] ~ dpois(lambda[i])
  }

  # Prior
  theta ~ dgamma(a, b)
}"
Y <- df$sum
```

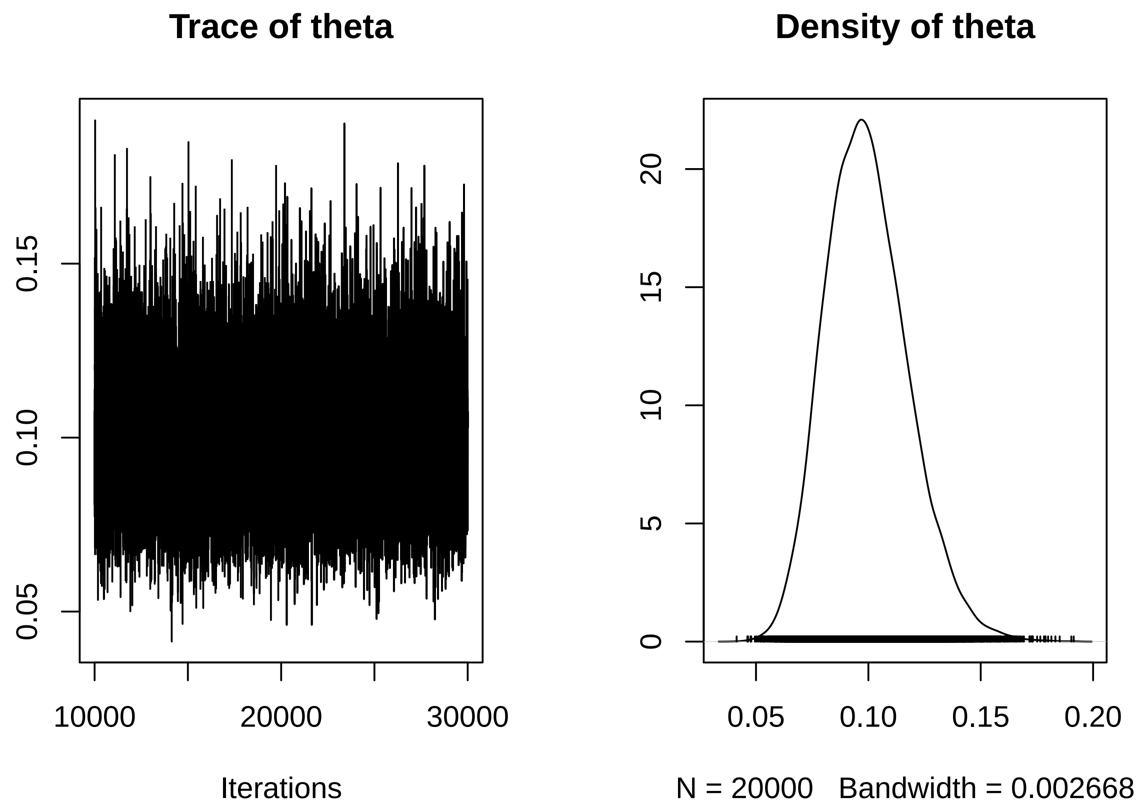
```r
model.concuss <- jags.model(textConnection(model_string), data = list(Y=Y,n=n,a=a,b=b))
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 32
##    Unobserved stochastic nodes: 33
##    Total graph size: 69
##
## Initializing model
```

```r
update(model.concuss, 10000, progress.bar="none"); # Burnin for 10000 samples

samp.theta <- coda.samples(model.concuss, variable.names=c("theta"),
        n.iter=20000, progress.bar="none")

summary(samp.theta)
```

```
##
## Iterations = 10001:30000
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 20000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##           Mean             SD       Naive SE Time-series SE
##      0.0995230      0.0183314      0.0001296      0.0001415
##
## 2. Quantiles for each variable:
##
##    2.5%     25%     50%     75%   97.5%
## 0.06757 0.08655 0.09819 0.11100 0.13876
```

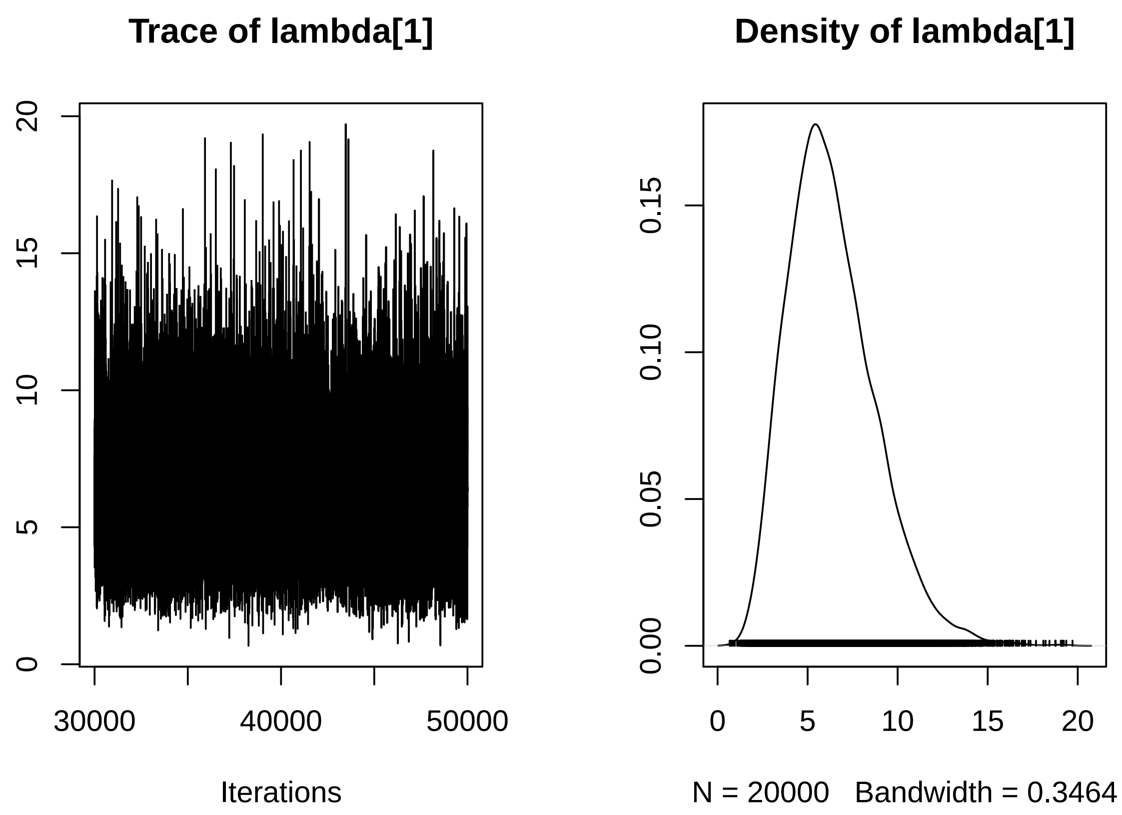
```r
plot(samp.theta)
```




```r
samp.lambda1 <- coda.samples(model.concuss, variable.names=c("lambda[1]"),
        n.iter=20000, progress.bar="none")

summary(samp.lambda1)
```

```
##
## Iterations = 30001:50000
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 20000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##           Mean             SD       Naive SE Time-series SE
##        6.35249        2.41543        0.01708        0.01708
##
## 2. Quantiles for each variable:
##
##   2.5%    25%    50%    75%  97.5%
##  2.545  4.610  6.055  7.784 11.808
```

```r
plot(samp.lambda1)
```




```r
mean(unlist(samp.lambda1))
```

```
## [1] 6.352491
```