Heirarchical Modeling of Photon Emission Count Data Using Non-Normally Distributed Hyperparameters

By Meenal Patel, Ali Ahmad Salahi, Ethan Bell, and Adam Kuhn

I. Problem & Motivation

Researchers are always looking for ways to reduce complexity in their models. Data, predictably, are not quite as willing to oblige this wish as researchers might want it to be. Practically, observations are often costly and difficult to obtain; obtaining photon counts in order to model the brightness of distant stars is one such example. At the time the technology was developed, single photon counters, the hardware used to collect photon count data, were prohibitively expensive even to most professional astronomers. The first such device was installed on the Hale telescope, first built in 1948 and at the time one of the costliest science expenditures in world history.

More important for our purposes is the fact that we must decide what assumptions to make about our data. Classically, the options were to fit a model to each individual observation, or to pool them all and fit a single model, assuming that each observation is drawn from an identical population. Bayesian hierarchical modeling is a “best of both worlds” approach, wherein we assume that the population parameters of a particular group (in this case, the group being the galaxy to which a particular star belongs) is itself drawn from a common distribution, that distribution’s parameters being governed by sub-distributions with known or inferred constant parameters. This approach allows us to partially pool information across groups, which is particularly useful when the number of groups is large and the sample size in each group ranges from small to very large.

II. Model

In this case we are interested in estimating the variability of star brightness across groups using photon count data. The model fit to the data distribution is a Poisson model,

$$Y_{ji} \vee \lambda_j \quad Pois(\lambda_j)$$

where λ_j is the mean of a particular group *j*. The means $\lambda_1, \ldots, \lambda_n$ are drawn from the distribution

$$\lambda_j \vee \alpha, \beta \quad Gamma(\alpha, \beta) \qquad \alpha \quad Gamma(a_\alpha, b_\alpha) \qquad \beta \quad Gamma(a_\beta, b_\beta)$$

where α, β control the distribution based upon the hyperparameters a_α, b_α , a_β, b_β , which are constants selected based upon prior knowledge (or lack thereof) about the distribution of group means. Our model is based upon a “naïve prior” approach, however other approaches were discussed before being discarded on the reasoning that they might import more assumptions to the model than necessary. The assumptions of the model are, we believe, reasonable:

(1) A Poisson model is the simplest probability distribution that best represents the nature of the data as discrete and with no upper bound,
(2) a conjugate Gamma prior is appropriate as it avoids the need for messy numerical integration and gives a closed-form expression for the posterior distribution, and
(3) Star luminosity is related significantly to the galaxy the star inhabits, which requires a deeper analysis than a mere least-squares linear regression.

Under these assumptions, our joint and conditional posterior distributions are:

$$\pi\left(\lambda_j, \alpha, \beta \middle| y_{ij}\right) \propto \left[\prod_{j=1}^{J} \lambda_j^{\alpha+\sum_{i=1}^{n_i} y_{ij}-1} e^{-\left(n_i+\beta\right)\lambda_j}\right] \frac{\alpha^{a_\alpha-1} e^{-b_\alpha \alpha}}{\left[\Gamma(\alpha)^J\right]} \beta^{J\alpha+a_\beta-1} e^{-b_\beta \beta}$$

$$\pi\left(\lambda_j \middle| \alpha, \beta, y_{ij} ¿\ Gamma\left(\alpha+\sum_{i=1}^{n_i} y_{ij}, n_i+\beta\right)\right.$$

$$\lambda_j$$

$$J\alpha+a_\beta, b_\beta+\sum_{j=1}^{J} ¿$$

$$\pi\left(\beta \middle| \lambda_j, \alpha, y_{ij}\right)\ Gamma ¿$$

$$\pi_{\alpha \vee \beta, \lambda_j, y_{ij}}\left(\alpha \middle| \beta, \lambda_j, y_{ij}\right) \propto \frac{\left(\sum_{j=1}^{J} \lambda_j\right)^{\alpha} \alpha^{a_\alpha-1} \beta^{J\alpha} e^{-b_\alpha \alpha}}{\left[\Gamma(\alpha)\right]^J}$$

III. Computational Details

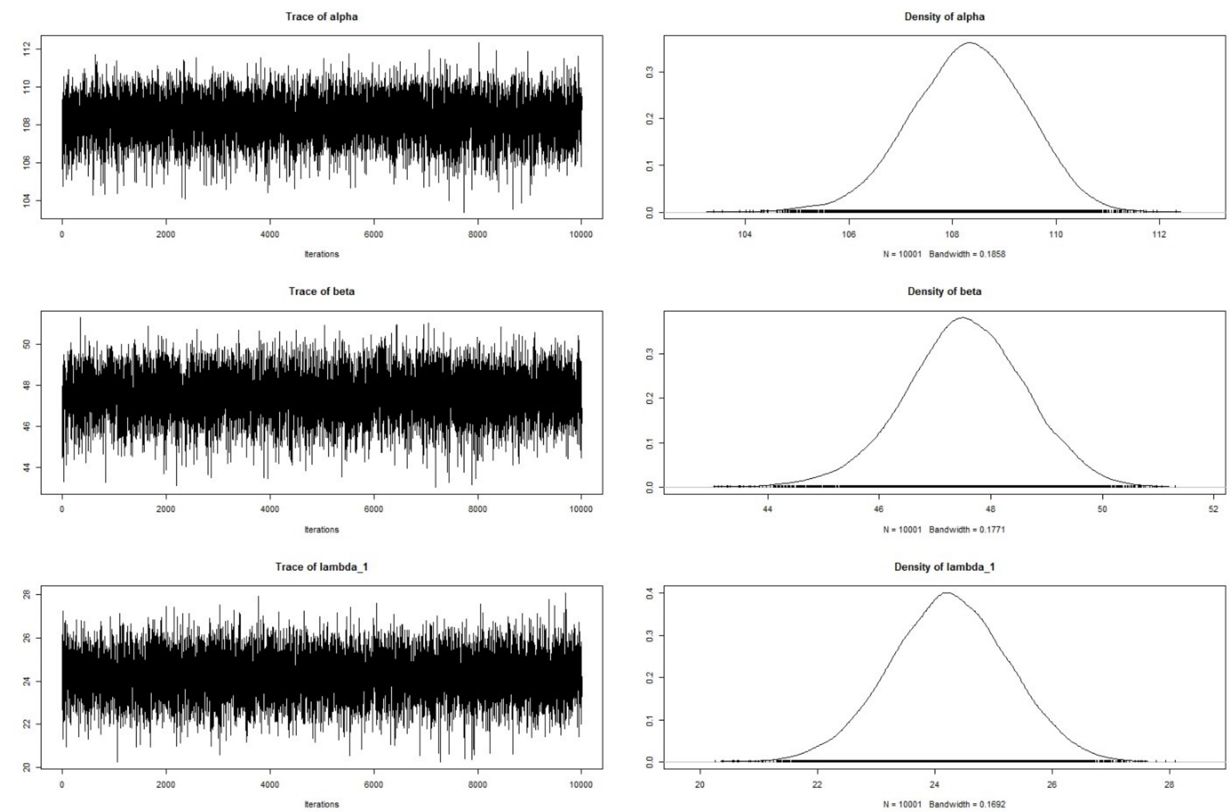
In order to sample from the posterior distribution, we employed a Gibbs sampling algorithm with a burn-in period of 10000 draws; iterative draws from each of the conditional posterior distributions were carried out, with $\pi_{\alpha \vee \beta, \lambda_j, y_{ij}}\left(\alpha \middle| \beta, \lambda_j, y_{ij}\right)$ modeled using an inverse-CDF approach. Using different starting points and hyperparameters, both very large, small, and based on the MLE's of the data, we returned very good consistent estimates over successive paths, hardly differing given the different starting points. This showed our model was not very sensitive and converged to practically the same estimates.

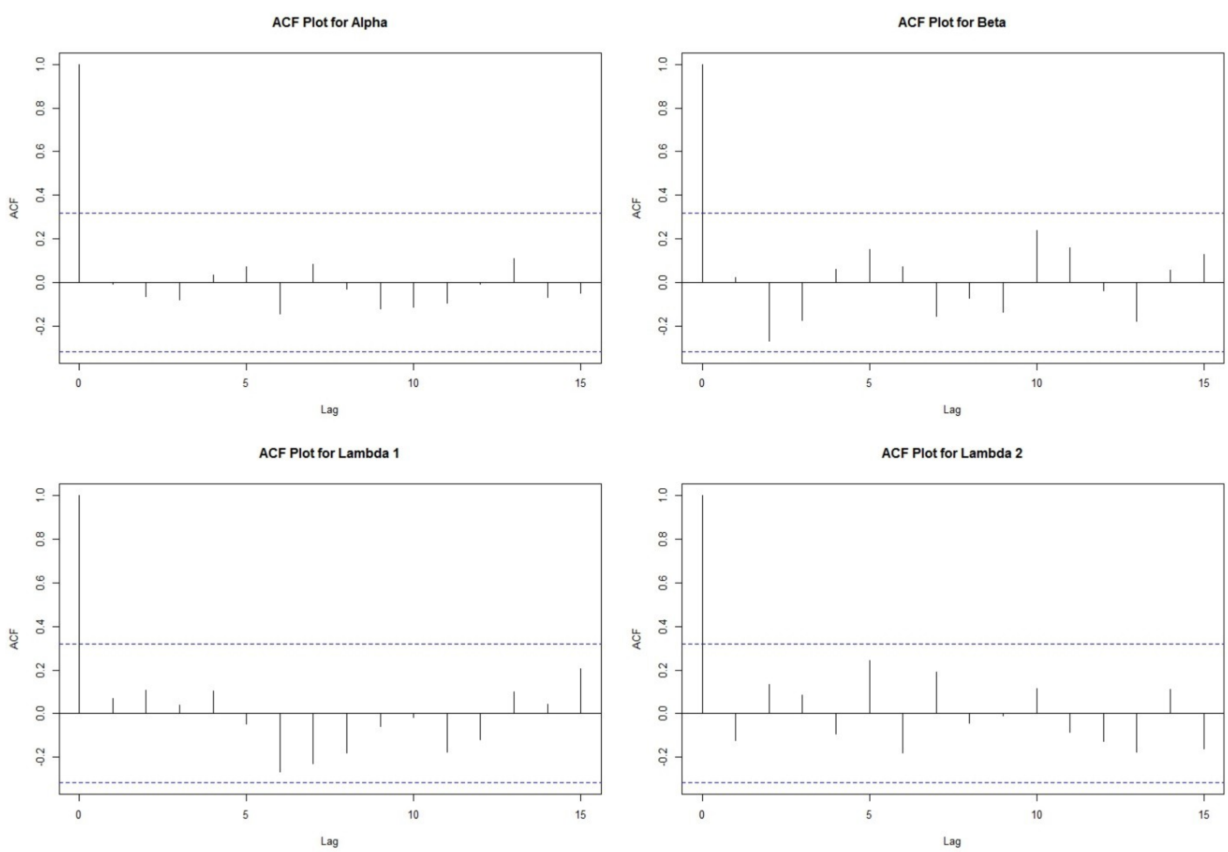
Our Gibbs Sampler gave us these estimates of medians, means, and 95% Credible Intervals based on the quantiles. As evidenced, our standard errors are all very low, and the variability between lambda groups is rather large, reflecting the range of group means in the data. Our credible intervals are also rather narrow, reflected in the standard error. This partial summary R output is included below:

```
Iterations = 1:1e+05
Thinning interval = 1
Number of chains = 1
Sample size per chain = 1e+05

1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:

              Mean     SD Naive SE Time-series SE
alpha     108.3226 1.1137 0.003522       0.003522
beta       47.5501 1.0896 0.003446       0.003446
lambda_1   24.2712 1.0106 0.003196       0.003196
lambda_3    5.9698 1.4186 0.004486       0.004486
lambda_4   12.5678 0.9355 0.002958       0.002958
lambda_11  32.0733 0.9360 0.002960       0.002960
lambda_15  21.7240 0.9264 0.002930       0.002930
lambda_16 197.6961 1.0816 0.003420       0.003444
lambda_19  11.7382 1.0050 0.003178       0.003178
lambda_20   3.7938 0.8456 0.002674       0.002659
lambda_22  77.4988 1.2092 0.003824       0.003824
lambda_24 117.4029 1.1210 0.003545       0.003545
lambda_25   7.4136 1.2601 0.003985       0.003985
lambda_27  99.7363 1.0681 0.003378       0.003378
lambda_28  77.9822 1.0669 0.003374       0.003374
lambda_29  41.9210 1.1013 0.003483       0.003483
lambda_36   0.5625 0.5527 0.001748       0.001748

2. Quantiles for each variable:

               2.5%      25%      50%      75%   97.5%
alpha     106.05396 107.5985 108.3544 109.0776 110.426
beta       45.34600  46.8353  47.5673  48.2841  49.649
lambda_1   22.28020  23.5958  24.2754  24.9488  26.248
lambda_3    3.54101   4.9523   5.8555   6.8559   9.042
lambda_4   10.76978  11.9306  12.5524  13.1867  14.443
lambda_11  30.22072  31.4508  32.0799  32.7018  33.894
lambda_15  19.90968  21.1034  21.7231  22.3416  23.547
lambda_16 195.48413 196.9947 197.7244 198.4257 199.748
lambda_19   9.81322  11.0511  11.7195  12.4023  13.753
lambda_20   2.33838   3.1947   3.7266   4.3215   5.636
lambda_22  75.02630  76.7172  77.5331  78.3162  79.786
lambda_24 115.10954 116.6767 117.4348 118.1617 119.522
lambda_25   5.14317   6.5309   7.3436   8.2178  10.062
lambda_27  97.56965  99.0388  99.7589 100.4638 101.764
lambda_28  75.80958  77.2897  78.0048  78.7008  80.013
lambda_29  39.70437  41.2013  41.9391  42.6567  44.051
lambda_36   0.01999   0.1685   0.3915   0.7799   2.040
```

Some diagnostic results of the posterior sampling are included below; in particular, the effective sample sizes for all parameters are quite good, and the trace and ACF plots on the next page show little to no dependency structure in the draws.

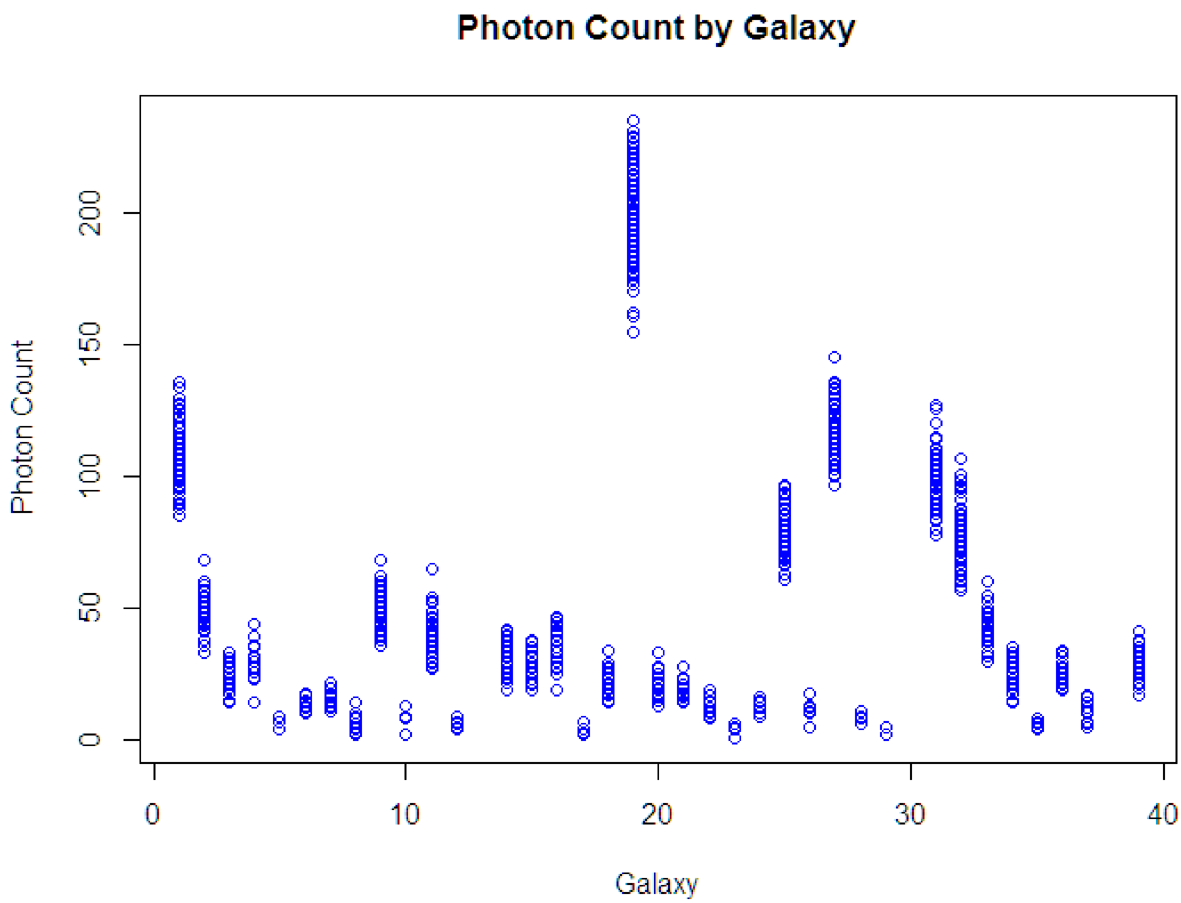
Parameter	**Effective Sample Size**
α	100000
β	100000
λ_6	98774 (smallest)
λ_{35}	102905 (largest)


Trace of alpha
Iterations
Density of alpha
N = 10001 Bandwidth = 0.1858
Trace of beta
Iterations
Density of beta
N = 10001 Bandwidth = 0.1771
Trace of lambda_1
Iterations
Density of lambda_1
N = 10001 Bandwidth = 0.1692


ACF Plot for Alpha
ACF
Lag
ACF Plot for Beta
ACF
Lag
ACF Plot for Lambda 1
ACF
Lag
ACF Plot for Lambda 2
ACF
Lag

IV. Data Analysis

The data used for the project was obtained from the course website and consists of photon counts of 1332 stars across a total of 36 galaxies (J=36). Y_{ji} is the number of photons counted from a particular star of index *i* in group *j*. Plotting the count data by group yields interesting results:



It's apparent from this plot that the variability of the data is correlated positively with the mean, which at least preliminarily affirms the selection of a Poisson model. Also apparent from the plot is the fact that means can differ significantly from galaxy to galaxy; this may be accounted for by the distance of a galaxy from the observer, or by true differences in the means themselves (as "not all galaxies are created equal", and some are quite a good deal older than others). It is interesting to note that older, more distant galaxies tend to have a more homogenous population of stars due to many of the larger stars having shorter lifespans and smaller stars not contributing as much to count data as main sequence stars, which would account for lowered variability in the data.

V. Lessons Learned

There are a number of factors to consider when embarking upon a Bayesian analysis, and not all of them are readily apparent at the outset. Only after completing the Gibbs sampling algorithm and noting that the range in the traceplots was perhaps a bit higher than desired, for instance, did we consider alternative approaches to sampling from $\pi\left(\alpha \middle| \beta, \lambda_j, y_{ij}\right)$ might yield better results than inverse-CDF sampling. Additionally, careful selection of values for the prior hyperparameters is challenging when we do not know much about the population from which the observations are being drawn; none of us are astronomers, for instance, and absent a specialist in the field to discuss it with, our second-best option was to begin with values that reflected our lack of knowledge, but choosing a prior with the assistance of an expert in the field might help our model to more accurately represent the population.

Code Appendix

```
#Bayesian Hierarchical Poisson Model Function:
library(lattice)
library(coda)

hlm <- function(nsamples, burnin, y, group.vector, a_alpha, b_alpha, a_beta, b_beta,
lambda.curr, alpha.curr, beta.curr)
{
  # Gibbs Sampling:
  "sample.lambda" <- function(nsamples, J, alpha, beta, sigma.y, n)
  {
    lambda_alpha = sapply(1:J, function(x) alpha + sigma.y[x])
    lambda_beta = sapply(1:J, function(x) n[x] + beta)
    return(sapply(1:J, function(x) rgamma(n = nsamples, lambda_alpha[x],
lambda_beta[x])))
  }
  "sample.alpha" <- function(nsamples, a_alpha, b_alpha)
  {
    u = runif(n=nsamples)
    return(qgamma(u, a_alpha, b_alpha))
  }
  "sample.beta" <- function(nsamples, alpha, beta, J, a_beta, b_beta, sigma.lambda)
  {
    beta_alpha = J*alpha + a_beta
    beta_beta = b_beta + sigma.lambda
    return(rgamma(n=nsamples, beta_alpha, beta_beta))
  }
  # Extract sufficient statistics needed for sampling
  N = length(y)
  group.levels = unique(group.vector)
  J = length(group.levels)
  n = rep(NA,J)
  sigma.y = rep(NA,J)
  sigma.lambda = sum(lambda.curr)
  for (i in 1:J){
    group.i <- as.logical(group.vector == group.levels[i])
    n[i] <- sum(group.i)
    sigma.y[i] <- sum(y[group.i])
  }
  # Storage for the MCMC draws:
  post.draws <- matrix(NA,nrow=nsamples,ncol=J+2)
  colnames(post.draws) <- c("alpha","beta",paste("lambda_",1:J,sep=""))
  # Gibbs sampler:
  for (i in 1:(nsamples+burnin))
  {
    lambda.curr <- sample.lambda(nsamples = 1, J = J, alpha = alpha.curr, beta =
beta.curr, sigma.y = sigma.y, n = n)
    alpha.curr <- sample.alpha(nsamples = 1,a_alpha =a_alpha, b_alpha = b_alpha)
    beta.curr <- sample.beta(nsamples = 1, alpha = alpha.curr, beta = beta.curr, J =J,
a_beta = a_beta, b_beta = b_beta, sigma.lambda = sigma.lambda)
    if (i > burnin){
      post.draws[i-burnin,] <- c(lambda.curr, alpha.curr, beta.curr)
    }
    if (i %% 1000 == 0)
      cat(paste("Finished iteration ",i,"...\n",sep=""))
  }
  # Reformat to MCMC draws:
  post.draws <- mcmc(post.draws)
  # Return:
  return(post.draws)
}

#Bayesian Hierarchical Poisson Sampling:

#Reading in the Data

data = read.table("C:/Users/Ali/Desktop/stars.txt", header=TRUE, quote="\"")
y = data$photon.count
galaxy = data$galaxy
```

```r
sapply(1:39, function(p) galaxy==p)
#Initial Plot of the Data
plot(y~galaxy, ylab = "Galaxy", main = "Photon Count by Galaxy", col = "blue")

# Prior for Alpha:
a_alpha = 1
b_alpha = 1

# Prior for Beta:
a_beta = 1
b_beta = 1

#Current State:
alpha.curr = 1
beta.curr = 1
lambda.curr = rep(mean(y)^2/var(y), length(unique(galaxy)))

#Number of Samples and Burnin Period
nsamples = 100000
burnin = 10000

lambdaMLE <- sapply(unique(galaxy), function(p) mean(data[c(which(p==galaxy)),2]))
alphabetaratio <- mean(lambdaMLE)

post.draws = hlm(nsamples = nsamples, burnin = burnin, y = y, group.vector = galaxy,
alphabetaratio, b_alpha, a_beta, b_beta, lambda.curr = lambda.curr, alpha.curr =
alphabetaratio, beta.curr = beta.curr)

post.drawsMLE = hlm(nsamples = nsamples, burnin = burnin, y = y, group.vector =
galaxy, 35, b_alpha, a_beta, b_beta, lambda.curr = lambdaMLE, alpha.curr = 35,
beta.curr = beta.curr)

#Sensitivity Analysis
summary(post.draws)
summary(post.drawsMLE)

# plot final 1000 samples:
plot(mcmc(post.draws[(nsamples-10000):nsamples,1:3]))


# Check diagnostics:

# Autocorrelation plots for Alpha, Beta, Lambda 1 and Lambda 2
lag.1.post.acf = apply(post.draws,2,acf,plot=FALSE)

par(mfrow = c(2,2))

lag.1.alpha = unlist(lapply(lag.1.post.acf,function(x){x[1][[1]][[1]][1]}))
lag.1.beta = unlist(lapply(lag.1.post.acf,function(x){x[2][[1]][[1]][1]}))
lag.1.lambda.1 = unlist(lapply(lag.1.post.acf,function(x){x[3][[1]][[1]][1]}))
lag.1.lambda.2 = unlist(lapply(lag.1.post.acf,function(x){x[4][[1]][[1]][1]}))

acf(lag.1.alpha, main = "ACF Plot for Alpha")
acf(lag.1.beta, main = "ACF Plot for Beta")
acf(lag.1.lambda.1, main = "ACF Plot for Lambda 1")
acf(lag.1.lambda.2, main = "ACF Plot for Lambda 2")


# Effective sample sizes:
ess = data.frame(effectiveSize(post.draws))
```