Fall 2019: MATH 347 Bayesian Statistics

```r
#install.packages("devtools")
require(devtools)
```

```
## Loading required package: devtools
## Loading required package: usethis
```

```r
devtools::install_github("bayesball/ProbBayes")
```

```
## Skipping install of 'ProbBayes' from a github remote, the SHA1 (805e6220) has not changed since last
##   Use `force = TRUE` to force installation
```

```r
require(ggplot2)
```

```
## Loading required package: ggplot2
```

```r
require(gridExtra)
```

```
## Loading required package: gridExtra
```

```r
require(ProbBayes)
```

```
## Loading required package: ProbBayes
## Loading required package: LearnBayes
## Loading required package: shiny
```

```r
require(tidyverse)
```

```
## Loading required package: tidyverse
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.1     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v lubridate 1.9.2     v tibble    3.2.1
## v purrr     1.0.1     v tidyr     1.3.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::combine() masks gridExtra::combine()
## x dplyr::filter()  masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```r
require(runjags)
```

```
## Loading required package: runjags
##
## Attaching package: 'runjags'
##
## The following object is masked from 'package:tidyr':
##
##     extract
```

```r
crcblue <- "#2905a1"

dramadata <- read.csv("KDramaData.csv", header = T)
```

```
KBSdrama = dramadata[dramadata$Producer==2,]
KBSdrama$Schedule = as.factor(KBSdrama$Schedule)
```

Lab 4: Hierarchical models for Kdrama rating

Author:________Shang-Chieh Wei ________

Total Grade for Lab 4: /18

Comments (optional)

Template for lab report

Instructions: This is the template you will use to type up your responses to the exercises. To produce a document that you can print out and turn in just click on Knit PDF above. All you need to do to complete the lab is to type up your BRIEF answers and the R code (when necessary) in the spaces provided below.

It is strongly recommended that you knit your document regularly (minimally after answering each exercise) for two reasons.

1. Ensure that there are no errors in your code that would prevent the document from knitting.
2. View the instructions and your answers in a more legible, attractive format.

```
# Any text BOTH preceded by a hashtag AND within the ```{r} ``` code chunk is a comment.
# R indicates a comment by turning the text green in the editor, and brown in the knitted
# document.
# Comments are not treated as a command to be interpreted by the computer.
# They normally (briefly!) describe the purpose of your command or chunk in plain English.
# However, for this class, they will have a different goal, as the text above and below
# each chunk should sufficiently describe the chunk's contents.
# For this class, comments will be used to indicate where your code should go, or to give
# hints for what the code should look like.
```

Overview

We have explored the Kdrama rating dataset with a hierarchical model. The sampling density, the two-stage prior distribution for μ_j's and the prior distribution for σ are presented below.

- The sampling density for group j, and $j = 1, \cdots, J$:

$$Y_{ij} \overset{i.i.d.}{\sim} \text{Normal}(\mu_j, \sigma), \tag{1}$$

  where $i = 1, \cdots, n_j$ and n_j is the number of observations in group j.

- The stage 1 prior distribution for μ_j :

$$\mu_j \sim \text{Normal}(\mu, \tau). \tag{2}$$

- The stage 2 prior distribution for μ_j:

$$\mu, \tau \sim \text{g}(\mu, \tau). \tag{3}$$

  Hyperpriors:

$$\begin{aligned} \mu \mid \mu_0, \gamma_0 &\sim \text{Normal}(\mu_0, \gamma_0), & (4)\\ 1/\tau^2 \mid \alpha_\tau, \beta_\tau &\sim \text{Gamma}(\alpha_\tau, \beta_\tau). & (5) \end{aligned}$$

- The prior distribution for σ:

$$1/\sigma^2 \sim \text{Gamma}(\alpha_\sigma, \beta_\sigma). \tag{6}$$

We have set $\mu_0 = 0.1, \gamma_0 = 0.5, \alpha_\tau = \beta_\tau = \alpha_\sigma = \beta_\sigma = 1$, and obtained posterior summaries below:

```
        Lower95 Median Upper95   Mean     SD Mode    MCerr MC%ofSD SSeff    AC.10 psrf
mu      -0.4905 0.1080   0.668 0.1047 0.2884   NA 0.004181     1.4  4758 -0.01052   NA
tau      0.3527 0.6585   1.250 0.7206 0.2749   NA 0.004198     1.5  4288 -0.01511   NA
mu_j[1] -0.0720 0.0713   0.217 0.0727 0.0724   NA 0.001024     1.4  5000  0.00223   NA
mu_j[2] -0.0569 0.0994   0.253 0.0993 0.0800   NA 0.001131     1.4  5000  0.01881   NA
mu_j[3] -0.2415 0.0448   0.349 0.0427 0.1511   NA 0.002073     1.4  5310 -0.00797   NA
mu_j[4] -0.0248 0.1914   0.399 0.1924 0.1075   NA 0.001520     1.4  5000 -0.01727   NA
sigma    0.2011 0.2616   0.333 0.2650 0.0346   NA 0.000554     1.6  3908  0.00988   NA
```

We have noticed the issues of negative draws of some parameter which should have been strictly non-negative, including `mu`, `mu_j[1]` through `mu_j[4]`, corresponding to μ and μ_1 through μ_4 in the model. Since these 5 parameters indicate the mean of the mean rating, and the means of ratings, they should be non-negative.

This lab is to explore different prior specifications that would prevent this from happening.

Truncated Normal distributions for the hyperprior for μ and priors for μ_j's

We know that μ and μ_j's should be non-negative. If we still want to use a hyperprior/prior distribution related to the Normal distribution, we can consider the truncated normal distribution.

The truncated Normal distribution

From Wikipedia: The truncated normal distribution is the probability distribution derived from that of a normally distributed random variable by bounding the random variable from either below or above (or both).

Suppose $Y \sim \text{Normal}(\mu, \sigma)$ has a Normal distribution and lies within the interval $Y \in (a, b)$, $-\infty \leq a < b \leq \infty$. Then Y conditional on $a < Y < b$ has a truncated Normal distribution, with pdf:

$$f(y \mid \mu, \sigma, a, b) = \frac{\phi(\frac{y-\mu}{\sigma})}{\sigma\left(\Phi(\frac{b-\mu}{\sigma}) - \Phi(\frac{a-\mu}{\sigma})\right)}, \tag{7}$$

where $\phi(.)$ is the pdf of the standard Normal distribution (i.e. $\text{Normal}(0, 1)$)and $\Phi(.)$ is the cdf (cumulative distribution function) of the standard Normal distribution.

Specifying a truncated Normal hyperprior/prior in JAGS

In the previous hierarchical model, where regular Normal prior distribution is assigned to μ_j, the syntax is:

```
for (j in 1:J){
mu_j[j] ~ dnorm(mu, invtau2)
}
```

If we want to use a truncated Normal prior distribution with only non-negative values of μ_j's, one can use the following syntax:

```
for (j in 1:J){
mu_j[j] ~ dnorm(mu, invtau2)T(0,)
}
```

```
modelString <- "
model {
## likelihood
for (i in 1:N){
y[i] ~ dnorm(mu_j[schedule[i]], invsigma2)
}

## priors
for (j in 1:J){
mu_j[j] ~ dnorm(mu, invtau2)T(0,)
}
invsigma2 ~ dgamma(a_g, b_g)
sigma <- sqrt(pow(invsigma2, -1))

## hyperpriors
mu ~ dnorm(mu0, 1/g0^2)T(0,)
invtau2 ~ dgamma(a_t, b_t)
tau <- sqrt(pow(invtau2, -1))
}
"
```

```
y = KBSdrama$Rating
schedule = KBSdrama$Schedule
N = length(y)
J = length(unique(schedule))

initsfunction <- function(chain){
  .RNG.seed <- c(1,2)[chain]
  .RNG.name <- c("base::Super-Duper",
               "base::Wichmann-Hill")[chain]
  return(list(.RNG.seed=.RNG.seed,
              .RNG.name=.RNG.name))
}

the_data <- list("y" = y, "schedule" = schedule, "N" = N, "J" = J,
                 "mu0" = 0.1, "g0" = 0.5,
                 "a_t" = 1, "b_t" = 1,
                 "a_g" = 1, "b_g" = 1)
```

```
posterior <- run.jags(modelString,
                      n.chains = 1,
                      data = the_data,
                      monitor = c("mu", "tau", "mu_j", "sigma"),
                      adapt = 1000,
                      burnin = 5000,
                      sample = 5000,
                      thin = 1,
                      inits = initsfunction)
```

Exercise 1: Give appropriate truncated Normal prior distribution for μ_j's and truncated Normal hyperprior distribution for μ. Run the new hierarchical model, and obtain the posterior summaries for all 7 parameters. Verify that the posterior draws of `mu` and `mu_j[1]` through

mu_j[4] are all non-negative. Include the 2-by-2 traceplot + cdf + historgram + ACF plot for mu_j[1] (Hint: use the plot(posterior, vars = "mu_j[1]" command). Comment on the MCMC diagnostics for mu_j[1].

```
## Loading required namespace: rjags
```

```
## Compiling rjags model...
## Calling the simulation using the rjags method...
## Adapting the model for 1000 iterations...
## Burning in the model for 5000 iterations...
## Running the model for 5000 iterations...
## Simulation complete
## Calculating summary statistics...
```

```
## Warning: Convergence cannot be assessed with only 1 chain
```

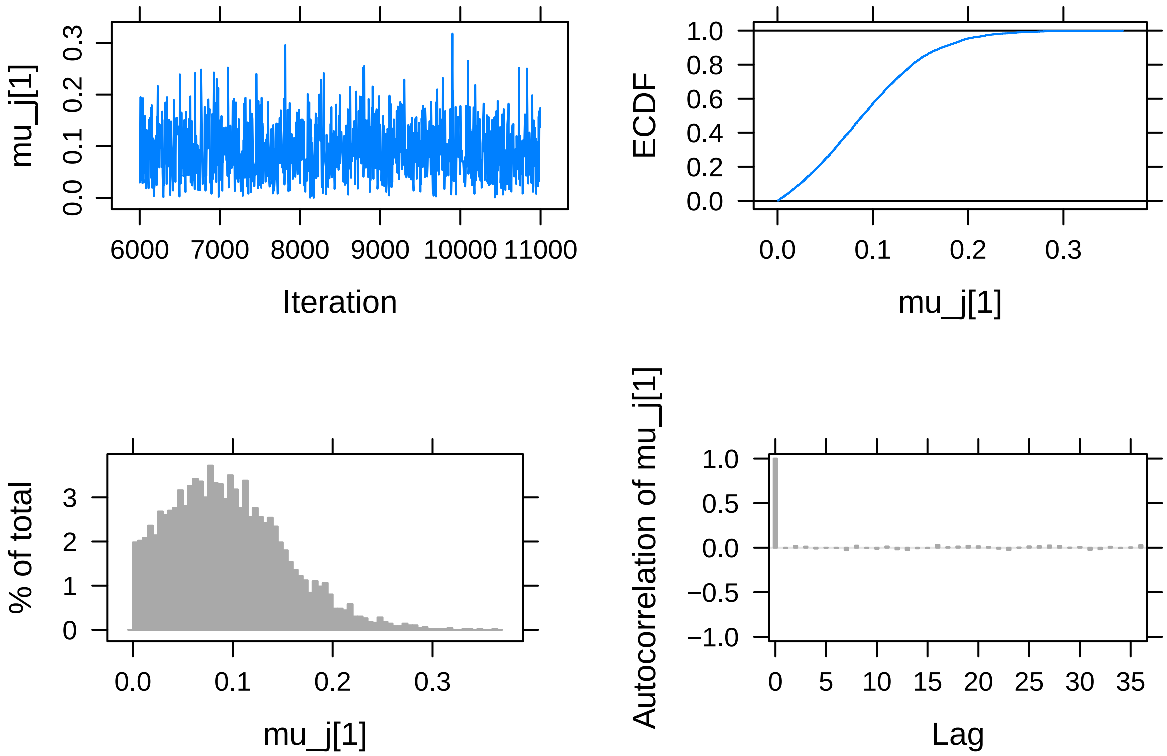
```
## Finished running the simulation
```

```
summary(posterior)
```

```
##              Lower95     Median   Upper95       Mean         SD Mode
## mu      0.0000418660 0.13647334 0.5034844 0.18079994 0.16139173   NA
## tau     0.3373797443 0.66070496 1.2708625 0.72654218 0.28497169   NA
## mu_j[1] 0.0001678117 0.08917457 0.1973330 0.09493157 0.05737621   NA
## mu_j[2] 0.0004592964 0.10914086 0.2409025 0.11605155 0.06735235   NA
## mu_j[3] 0.0000167566 0.11756044 0.3316200 0.13799521 0.10111323   NA
## mu_j[4] 0.0066445403 0.19918933 0.3723056 0.20289096 0.09712466   NA
## sigma   0.2017483176 0.25847935 0.3306072 0.26240868 0.03366969   NA
##                MCerr MC%ofSD SSeff        AC.10 psrf
## mu      0.0045894802     2.8  1237 -0.011833062   NA
## tau     0.0056666140     2.0  2529 -0.012405050   NA
## mu_j[1] 0.0008114222     1.4  5000 -0.012160235   NA
## mu_j[2] 0.0009525061     1.4  5000 -0.016693840   NA
## mu_j[3] 0.0015230764     1.5  4407  0.008611642   NA
## mu_j[4] 0.0013735501     1.4  5000 -0.001271426   NA
## sigma   0.0005055394     1.5  4436 -0.006560974   NA
```

```
plot(posterior, vars = "mu_j[1]")
```

```
## Generating plots...
```



Grade for Exercise 1: /6

Comments:

Log-normal distributions for the hyperprior for μ and priors for μ_j's

In addition to truncated Normal distribution, we can also consider the log-normal distribution.

The log-normal distribution

From Wikipedia: a log-normal (or lognormal) distribution is a continuous probability distribution of a random variable whose logarithm is Normally distributed. Thus, if the random variable Y is log-normally distributed, then $Y' = \ln(Y)$ has a Normal distribution.

A random variable which is log-normally distributed takes only positive real values, an appealing feature for μ and μ_j's in the Kdrama rating application.

If Y $\sim$ Normal(μ, σ), its pdf is:

$$f(y \mid \mu, \sigma) = \frac{1}{y} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(\ln(y) - \mu)^2}{2\sigma^2}\right). \tag{8}$$

Specifying a log-normal hyperprior/prior in JAGS

In the previous hierarchical model, where regular Normal prior distribution is assigned to μ_j, the syntax is:

```
for (j in 1:J){
mu_j[j] ~ dnorm(mu, invtau2)
}
```

If we want to use a log-normal prior distribution with only non-negative values of μ_j's, one can use the following syntax:

```
for (j in 1:J){
mu_j[j] ~ dlnorm(mu, invtau2)
}
```

```
modelString <- "
model {
  ## likelihood
  for (i in 1:N) {
    y[i] ~ dnorm(mu_j[schedule[i]], invsigma2)
  }

  ## priors
  for (j in 1:J) {
    mu_j[j] ~ dlnorm(mu, invtau2)
  }
  invsigma2 ~ dgamma(a_g, b_g)
  sigma <- sqrt(pow(invsigma2, -1))

  ## hyperpriors
  mu ~ dlnorm(mu0, 1/g0^2)
  invtau2 ~ dgamma(a_t, b_t)
  tau <- sqrt(pow(invtau2, -1))
}
"
```

```
y = KBSdrama$Rating
schedule = KBSdrama$Schedule
N = length(y)
J = length(unique(schedule))

initsfunction <- function(chain){
  .RNG.seed <- c(1,2)[chain]
  .RNG.name <- c("base::Super-Duper",
              "base::Wichmann-Hill")[chain]
  return(list(.RNG.seed=.RNG.seed,
              .RNG.name=.RNG.name))
}

the_data <- list("y" = y, "schedule" = schedule, "N" = N, "J" = J,
                 "mu0" = 0.1, "g0" = 0.5,
                 "a_t" = 1, "b_t" = 1,
                 "a_g" = 1, "b_g" = 1)
```

```
posterior <- run.jags(modelString,
                      n.chains = 1,
```

```
                      data = the_data,
                      monitor = c("mu", "tau", "mu_j", "sigma"),
                      adapt = 1000,
                      burnin = 5000,
                      sample = 5000,
                      thin = 1,
                      inits = initsfunction)
```

Exercise 2: Give appropriate log-normal prior distribution for μ_j's and log-normal hyperprior distribution for μ. Run the new hierarchical model, and obtain the posterior summaries for all 7 parameters. Verify that the posterior draws of `mu` and `mu_j[1]` through `mu_j[4]` are all non-negative. Include the 2-by-2 traceplot + cdf + historgram + ACF plot for `mu_j[1]` (Hint: use the `plot(posterior, vars = "mu_j[1]"` command). Comment on the MCMC diagnostics for `mu_j[1]`.

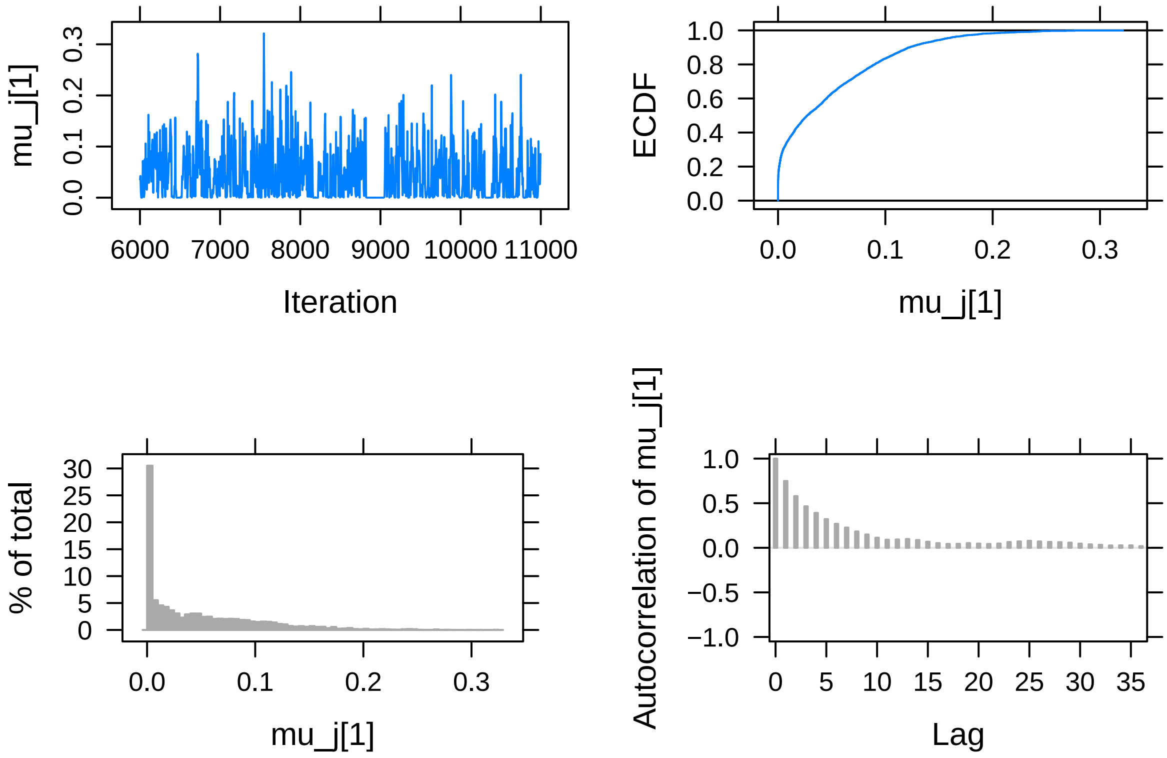
```
## Compiling rjags model...
## Calling the simulation using the rjags method...
## Adapting the model for 1000 iterations...
## Burning in the model for 5000 iterations...
## Running the model for 5000 iterations...
## Simulation complete
## Calculating summary statistics...

## Warning: Convergence cannot be assessed with only 1 chain

## Finished running the simulation
```

```
summary(posterior)
```

```
##               Lower95     Median    Upper95       Mean         SD Mode
## mu       2.541597e-01 0.84932703  1.8556657 0.94324168 0.46251708   NA
## tau      1.439095e+00 4.56047661 13.2946439 5.73252078 3.88459228   NA
## mu_j[1]  4.302811e-13 0.02711841  0.1546977 0.04702418 0.05342850   NA
## mu_j[2]  1.344185e-08 0.04444035  0.1851709 0.06312083 0.06277001   NA
## mu_j[3]  4.845379e-16 0.01867753  0.2273520 0.05463500 0.08074926   NA
## mu_j[4]  2.876256e-16 0.11779737  0.3217777 0.12897713 0.10577856   NA
## sigma    2.022892e-01 0.25908592  0.3301510 0.26220175 0.03364110   NA
##                 MCerr MC%ofSD SSeff       AC.10 psrf
## mu       0.0117021251     2.5  1562 -0.02078790   NA
## tau      0.4848641818    12.5    64  0.52604439   NA
## mu_j[1]  0.0024065417     4.5   493  0.11327974   NA
## mu_j[2]  0.0023791957     3.8   696  0.11664484   NA
## mu_j[3]  0.0042954001     5.3   353  0.27144970   NA
## mu_j[4]  0.0044502441     4.2   565  0.12915769   NA
## sigma    0.0005608731     1.7  3598  0.02194147   NA
```

```
plot(posterior, vars = "mu_j[1]")
```

```
## Generating plots...
```



Grade for Exercise 2: /6

Comments: Need to increase 'thin'

Your choice of distribution for the hyperprior for μ and priors for μ_j's

Exercise 3: **Give appropriate prior distribution for μ_j's and hyperprior distribution for μ of your own choosing. Run the new hierarchical model, and obtain the posterior summaries for all 7 parameters. Verify that the posterior draws of `mu` and `mu_j[1]` through `mu_j[4]` are all non-negative. Include the 2-by-2 traceplot + cdf + historgram + ACF plot for `mu_j[1]` (Hint: use the `plot(posterior, vars = "mu_j[1]"` command). Comment on the MCMC diagnostics for `mu_j[1]`.**

Grade for Exercise 3: /6

Comments: