Homework 4

Ryan Hastings
Sta 360: Bayesian and Modern Statistics
Duke University

July 27, 2020

Marriage Data (Hoff 7.4)

The file `agehw.dat` contains data on the ages of 100 married couples sampled from the U.S. population.

```r
agehw <- read.table(
  url("http://www2.stat.duke.edu/~pdh10/FCBS/Exercises/agehw.dat"),
  header = T
)
```

Part A

Before you look at the data, use your own knowledge to formulate a semiconjugate prior distribution for $\theta = (\theta_h, \theta_w)^T$ and Σ, where θ_h, θ_w are mean husband and wife ages, and Σ is the covariance matrix.

I will use a multivariate normal model $Y \sim \mathcal{N}_2(\theta, \Sigma)$ with priors $\pi(\theta) = \mathcal{N}_2(\mu_0, \Lambda_0)$ and $\pi(\Sigma) = \mathcal{IW}_2(\nu_0, S_0)$

The mean age of people who are in married couples is likely between age 45 and 65, and the mean age for men and women will likely be similar. Setting the 95% CI to be (45, 65), the standard deviation would be 5, so the variance would be 25. Couples' ages are likely highly correlated, so I'll set the correlation to be 0.75. The off-diagonals of Λ_0 will be 0.75 * 25 = 18.75.

```r
mu_0 <- c(55, 55)
Lambda_0 <- matrix(c(25, 18.75, 18.75, 25), nrow = 2)
```

Most people who are part of a married couple will be between ages 20 and 90. Setting the 95% CI to be (20, 90), the standard deviation would be 17.5, so the variance would be about 306. Keeping the correlation at 0.75, the off-diagonals of S_0 will be 0.75 * 306 = 229.5. I am somewhat confident in this prior, so I will set $\nu_0 = 13$.

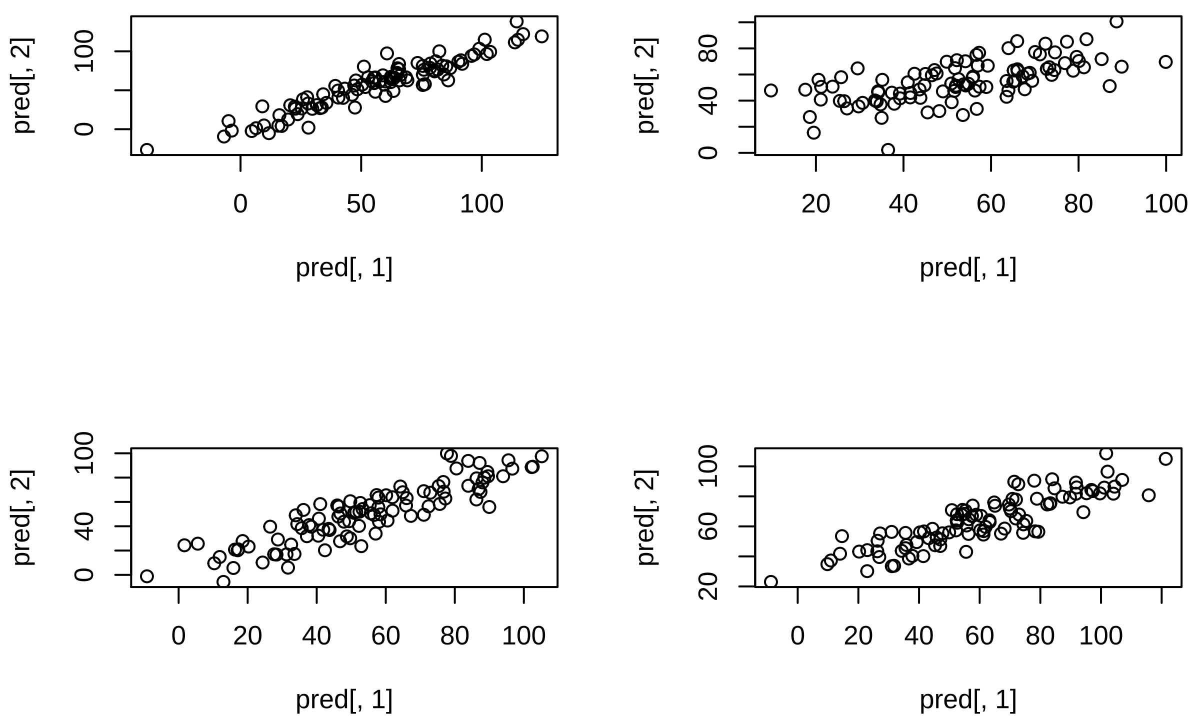
```r
nu_0 <- 13
S_0 <- (nu_0 - 2 - 1) * matrix(c(306, 229.5, 229.5, 306), nrow = 2)
```

Part B

Generate a prior predictive dataset of size $n = 100$, by sampling (θ, Σ) from your prior distribution and then simulating $Y_1, ..., Y_n \sim$ i.i.d. multivariate normal (θ, Σ). Generate several such datasets, make bivariate scatterplots for each dataset, and make sure they roughly represent your prior beliefs about what such a dataset would actually look like. If your prior predictive datasets do not conform to your beliefs, go back to part A and formulate a new prior. Report the prior that you eventually decide upon, and provide scatterplots for at least three prior predictive datasets.

I use my prior from part A.

```r
set.seed(4233)

par(mfrow=c(2,2))

for (i in 1:4) {
  theta <- mvrnorm(1, mu_0, Lambda_0)
  sigma <- riwish(nu_0, S_0)
  pred <- mvrnorm(100, theta, sigma)

  plot(pred[,1], pred[,2])
}
```



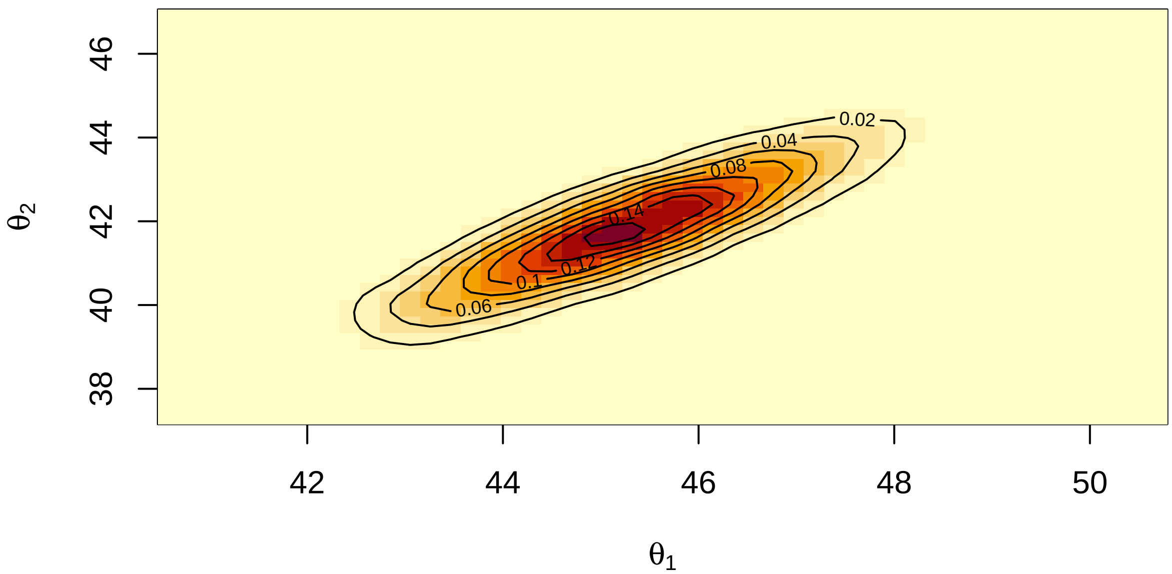
Part C

Using your prior distribution and the 100 values in the dataset, obtain an MCMC approximation to $p(\theta, \Sigma | y_1, ..., y_{100})$. Plot the joint posterior distribution of θ_h and θ_w, and also the marginal posterior density of the correlation between Y_h and Y_w, the ages of a husband and wife. Obtain 95% posterior confidence intervals for θ_h, θ_w and the correlation coefficient.

```r
Y <- agehw
n <- nrow(Y)
ybar <- apply(Y,2,mean)
Sigma <- cov(Y)

THETA <- SIGMA <- NULL
n_iter <- 10000; burn_in <- 0.3*n_iter
set.seed(34235)

for (s in 1:(n_iter+burn_in)){
  #update theta
  Lambda_n <- solve(solve(Lambda_0) + n*solve(Sigma))
  mu_n <- Lambda_n %*% (solve(Lambda_0) %*% mu_0 + n*solve(Sigma) %*% ybar)
  theta <- rmvnorm(1,mu_n,Lambda_n)

  #update Sigma
  S_theta <- (t(Y)-c(theta)) %*% t(t(Y)-c(theta))
  S_n <- S_0 + S_theta
  nu_n <- nu_0 + n
  Sigma <- riwish(nu_n, S_n)

  #save results past burn-in
  if(s > burn_in){
    THETA <- rbind(THETA,theta)
    SIGMA <- rbind(SIGMA,c(Sigma))
  }
}
colnames(THETA) <- c("theta_h","theta_w")
colnames(SIGMA) <- c("sigma_11","sigma_12","sigma_21","sigma_22")
```

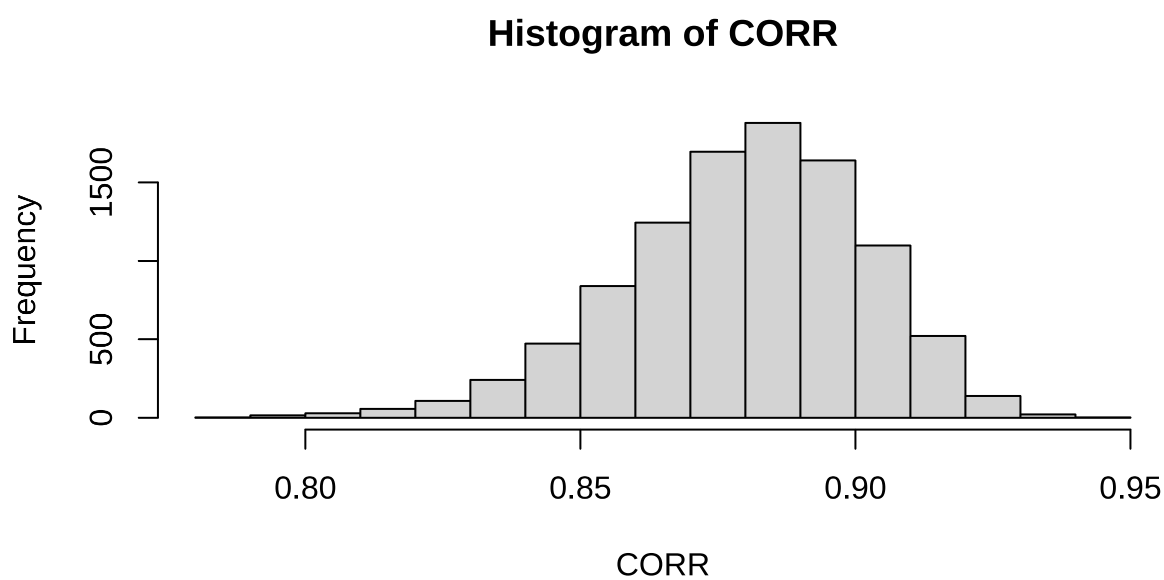
Plot of joint posterior distribution:

```r
theta.kde <- kde2d(THETA[,1], THETA[,2], n = 50)
image(theta.kde,xlab=expression(theta[1]),ylab=expression(theta[2]))
contour(theta.kde, add = T)
```



Plot of the marginal posterior density of the correlation between Y_h and Y_w:

```
CORR <- SIGMA[,2]/ (sqrt(SIGMA[,1]) * sqrt(SIGMA[,4]))

hist(CORR)
```



Confidence intervals:

```r
quantile(THETA[, "theta_h"], probs = c(0.025, 0.975))
```

```
##     2.5%    97.5%
## 42.64417 47.90244
```

```r
quantile(THETA[, "theta_w"], probs = c(0.025, 0.975))
```

```
##     2.5%    97.5%
## 39.27908 44.34442
```

```r
quantile(CORR, probs = c(0.025, 0.975))
```

```
##      2.5%     97.5%
## 0.8319848 0.9173860
```

The 95% posterior confidence interval for θ_h is about 42.64 to 47.90. The 95% posterior confidence interval for θ_w is about 39.28 to 44.34. The 95% posterior confidence interval for the correlation coefficient is about 0.832 to 0.917.

Part D (i)

Obtain 95% posterior confidence intervals for θ_h, θ_w and the correlation coefficient using the Jeffreys' prior distribution.

```r
Y <- as.matrix(agehw, ncol = 2)
n <- nrow(Y)
ybar <- apply(Y,2,mean)
Sigma <- cov(Y)

THETA_J <- SIGMA_J <- NULL
n_iter <- 10000; burn_in <- 0.3*n_iter
set.seed(34235)

for (s in 1:(n_iter+burn_in)){
  #update theta
  theta <- rmvnorm(1, ybar, Sigma/n)

  #update Sigma
  Sigma <- riwish(n + 1, (t(Y) - c(theta)) %*% t(t(Y) - c(theta)))

  #save results past burn-in
  if(s > burn_in){
    THETA_J <- rbind(THETA_J,theta)
    SIGMA_J <- rbind(SIGMA_J,c(Sigma))
  }
}
colnames(THETA_J) <- c("theta_h","theta_w")
colnames(SIGMA_J) <- c("sigma_11","sigma_12","sigma_21","sigma_22")
```

```r
quantile(THETA_J[, "theta_h"], probs = c(0.025, 0.975))
```

```
##     2.5%    97.5%
## 41.71702 47.08636
```

```r
quantile(THETA_J[, "theta_w"], probs = c(0.025, 0.975))
```

```
##     2.5%    97.5%
## 38.31899 43.41873
```

```r
CORR <- SIGMA_J[,2]/ (sqrt(SIGMA_J[,1]) * sqrt(SIGMA_J[,4]))
quantile(CORR, probs = c(0.025, 0.975))
```

```
##      2.5%     97.5%
## 0.8608245 0.9349164
```

The 95% posterior confidence interval for θ_h is about 41.72 to 47.09. The 95% posterior confidence interval for θ_w is about 38.32 to 43.42. The 95% posterior confidence interval for the correlation coefficient is about 0.861 to 0.935.

Part E

Compare the confidence intervals from Part D to those obtained in Part C. Discuss whether or not you think that your prior information is helpful in estimating θ and Σ, or if you think one of the alternatives in Part D is preferable. What about if the sample size were much smaller, say $n = 25$?

Both the lower bounds and upper bounds of the confidence intervals for θ_h and θ_w in Part C are higher than their counterparts in Part D. I set my prior for both to be centered at 55, while the data had a significantly lower sample mean. Therefore, the estimates in Part C were pulled toward this higher prior, and those in Part D were not.

Similarly, the upper and lower bounds of the interval for the correlation coefficient in Part C are lower than than the respective bounds in Part D. I set the prior correlation to be 0.75, but the correlation in the sample was higher. The results in Part C were pulled toward my prior guess.

For large data, the Jeffreys' prior might be preferable to setting a prior, as you have to specify a lot of hyperparameters about which you have little information and which ultimately do not affect the estimates all that much. For data with fewer than 25 observations, a carefully-chosen prior would likely be preferable to provide context to the small sample.

Question 2

Set Up

Simulate data assuming $y_i = (y_{i1}, y_{i2})^T \sim \mathcal{N}_2(\theta, \Sigma)$, $i = 1, \ldots, 100$, with $\theta = (0, 0)^T$ and Σ chosen so that the marginal variances are 1 and correlation $\rho = 0.8$.

```r
theta_0 <- c(0, 0)
sigma_0 <- matrix(c(1, 0.8, 0.8, 1), nrow = 2)

set.seed(39281)

Y <- rmvnorm(100, theta_0, sigma_0)
```

Assuming independent normal & inverse-Wishart priors for θ and Σ, that is, $\pi(\theta, \Sigma) = \pi(\theta)\pi(\Sigma)$, run Gibbs sampler (hyperparameters up to you but you must justify your choices) to generate posterior samples for (θ, Σ).

Setting hyperparameters for priors:

I assume data is centered around (0, 0), and the true mean is probably within (-1, 1). Therefore, I set the standard deviation to be 0.5 and the variance to be 0.25. I set prior correlation to 0.5, since I figure that they might be correlated but I am not sure how much so.

```r
mu_0 <- c(0, 0)
Lambda_0 <- matrix(c(0.25, 0.125, 0.125, 0.25), nrow=2)
```

95% of the data is probably within (-2, 2). Therefore I set the standard deviation to be 1 and the variance to be 1. I keep the correlation at 0.5, but I am not at all confident about this prior value, so I set ν_0 to be 4.

```r
nu_0 <- 4
S_0 <- matrix(c(1, 0.5, 0.5, 1), nrow=2)
```

Gibbs sampler:

```r
#Data summaries
n <- nrow(Y)
ybar <- apply(Y,2,mean)

#Initial values for Gibbs sampler
#No need to set initial value for theta, we can simply sample it first
Sigma <- cov(Y)

#Set null matrices to save samples
THETA <- SIGMA <- NULL

#set number of iterations and burn-in, then set seed
n_iter <- 10000; burn_in <- 0.3*n_iter
set.seed(3204)

for (s in 1:(n_iter+burn_in)){

  #update theta using its full conditional
  Lambda_n <- solve(solve(Lambda_0) + n*solve(Sigma))
  mu_n <- Lambda_n %*% (solve(Lambda_0)%*%mu_0 + n*solve(Sigma)%*%ybar)
  theta <- rmvnorm(1,mu_n,Lambda_n)

  #update Sigma
  S_theta <- (t(Y)-c(theta))%*%t(t(Y)-c(theta))
  S_n <- S_0 + S_theta
```

```
  nu_n <- nu_0 + n
  Sigma <- riwish(nu_n, S_n)

  #save results only past burn-in
  if(s > burn_in){
    THETA <- rbind(THETA,theta)
    SIGMA <- rbind(SIGMA,c(Sigma))
  }
}

colnames(THETA) <- c("theta_1","theta_2")
colnames(SIGMA) <- c("sigma_11","sigma_12","sigma_21","sigma_22")
```

Now, generate 50 new "test" data from the same sampling distribution, that is, $y_i^* = (y_{i,1}^*, y_{i,2}^*)^T \sim \mathcal{N}_2(\theta, \Sigma)$, $i = 1, \ldots, 50$. Keep the $y_{i,2}^*$ values but set the $y_{i,1}^*$ values to`NA` (make sure to save the "true" values somewhere!).

```
set.seed(8382)

y_star <- rmvnorm(50, theta_0, sigma_0)
y_test <- y_star[,2]
```

Using the posterior samples for (θ, Σ), based on the 100 "train" data, answer the following questions:

Part A

Generate predictive samples of $y_{i,1}^*$ given each $y_{i,2}^*$ value you kept, for the 50 test subjects. Show your sampler.

I use the formula for the conditional distribution of a multivariate normal model: $(y_{i,1} | y_{i,2}, \theta, \Sigma)$

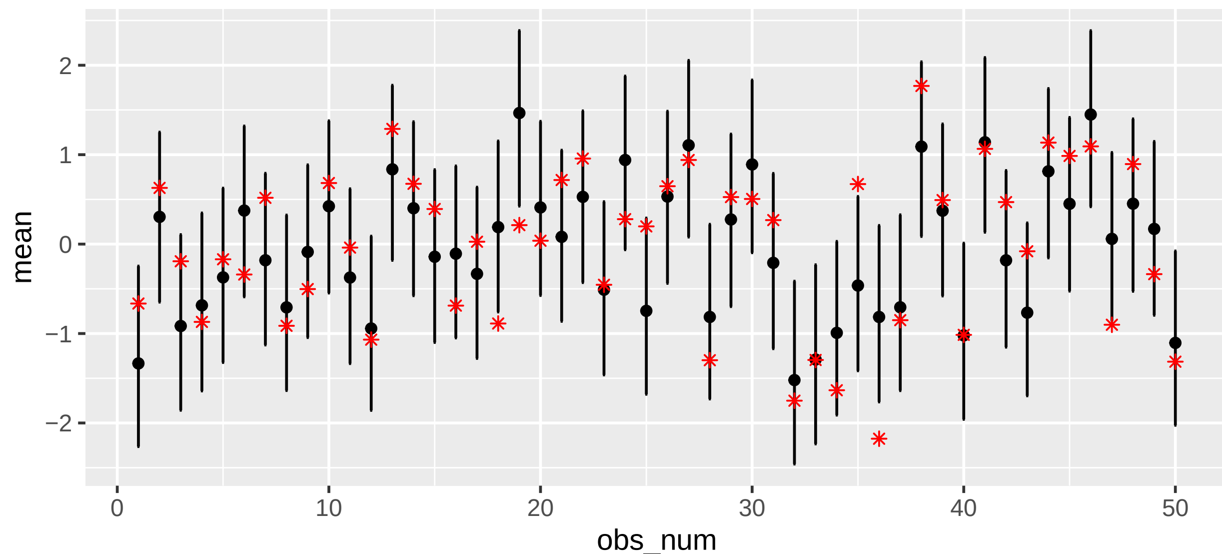
```
PRED <- NULL

set.seed(4843)

for (i in 1:50) {
  pred_distr <- rnorm(10000,
        THETA[,1] + SIGMA[,2]/sqrt(SIGMA[,4]) * (y_test[i] - THETA[,2]),
        sqrt(SIGMA[,1]) - SIGMA[,2]^2/sqrt(SIGMA[,4]))
  PRED <- rbind(PRED, pred_distr)
}
```

Part B

Using the samples from the predictive density obtained above, obtain $\mathbb{E}[y_{i,1}^* | y_{i,2}^*]$ for each of the test subjects, as well as a 95% posterior predictive interval. Make a plot containing all the intervals for each of the 50 subjects. In the plot, indicate where each $\mathbb{E}[y_{i,1}^* | y_{i,2}^*]$ falls within each interval.

```
CI <- NULL

for (i in 1:50) {
```

```
  exp <- mean(PRED[i,], na.rm = T)
  ci <- quantile(PRED[i,], prob = c(0.025, 0.975), na.rm = T)
  CI <- rbind(CI, c(i, exp, ci))
}

colnames(CI) <- c("obs_num", "mean","lower", "upper")

CI_df <- as_tibble(CI)

ggplot(CI_df, aes(x = obs_num)) +
  geom_errorbar(aes(ymin=lower, ymax=upper), width=.1) +
  geom_point(aes(y = mean)) +
  geom_point(aes(y = y_star[,1]), color = "red", shape = 8)
```



The expected values are marked by the black dots, while the true values are marked by the red asterisks.

Part C

What is the coverage of the 95% predictive intervals out of sample? That is, how many of the 95% predictive intervals contain the true $y^*_{i,1}$ values?

```
sum(y_star[,1] > CI_df$lower & y_star[,1] < CI_df$upper)
```

```
## [1] 45
```

45 of the 50 predictive intervals contain the true $y^*_{i,1}$ values.

Question 3

Suppose data consist of reaction times y_{ij} for subjects $i = 1, \ldots, n_j$ in experimental conditions $j = 1, \ldots, J$. Researchers inform you that it is reasonable to assume that reaction times follow an exponential distribution.

Part A

Describe a Bayesian hierarchical model for borrowing information across experimental conditions. Specify priors that will allow you to borrow information across the J conditions.

$$\begin{aligned} y_{ij}|\theta_j &\sim Exp(\theta_j) \\ \theta_j|a,b &\sim Ga(a,b) \\ \pi(b) &= Ga(\alpha,\beta) \\ \pi(a) &\propto e^{-\gamma a} \end{aligned}$$

Part B

Derive the Gibbs sampling algorithm for fitting your hierarchical model. What are the full conditionals?

Full conditional for θ_j:

$$\begin{aligned} \pi(\theta_j|\theta\_j, a, b, Y) &\propto \{\Pi_{i=1}^{n_j} p(y_{ij}|\theta_j)\} * p(\theta_j|a,b) \\ &\propto Gamma(a+n_j, b+\sum_{i=1}^{n_j} y_{ij}) \end{aligned}$$

Full conditional for b:

$$\begin{aligned} \pi(b|a,\theta_{1:J},Y) &\propto \{\Pi_{j=1}^{J} p(\theta_j|a,b)\} * \pi(b) \\ &\propto \{\Pi_{j=1}^{J} b^a e^{-b\theta_j}\} * b^{\alpha-1} e^{-\beta b} \\ &\propto b^{Ja} e^{-b\sum\theta_j} * b^{\alpha-1} e^{-\beta b} \\ &\propto Gamma(b; \alpha+Ja, \beta+\sum_{j=1}^{J}\theta_j) \end{aligned}$$

Full conditional for a:

$$\begin{aligned} \pi(a|b,\theta_{1:J},Y) &\propto \{\Pi_{j=1}^{J} p(\theta_j|a,b)\} * \pi(a) \\ &\propto \{\Pi_{j=1}^{J} \frac{b^a}{\Gamma(a)} \theta_j^{a-1}\} * e^{-\gamma a} \\ &\propto [\frac{b^a}{\Gamma(a)}]^J \{\Pi_{j=1}^{J} \theta_j^{a-1}\} * e^{-\gamma a} \end{aligned}$$

$$\ln\pi(a|b,\theta_{1:J},Y) \propto Ja\ln b - J\ln[\Gamma(a)] + (a-1)(\sum_{j=1}^{J}\ln\theta_j) - \gamma\alpha$$

```
hier_expo_sampler <- function (J, n_j, sum_y_j, alpha, beta, gamma) {

  # data summaries: J, n_j, sum_y_j
  # hyperparameters: alpha, beta, gamma

  # grid values for sampling a
  a_grid <- 1:5000
```

```r
  # initial values for Gibbs sampler
  a <- 1
  b <- 2

  #set number of iterations and burn-in and set seed
  n_iter <- 10000; burn_in <- 0.3*n_iter
  set.seed(4392)

  #set null matrices to save samples
  SAMPLES <- NULL

  # Gibbs Sampler
  for(s in 1:(n_iter+burn_in)){

    # update theta_j's
    theta_j <- rgamma(J, a + n_j, b + sum_y_j)

    # update b
    b <- rgamma(1, alpha + J*a, beta + sum(theta_j))

    # update a
    log_pi_a <- J*a_grid*log(b) - J*lgamma(a_grid) +
      (a_grid-1)*sum(log(theta_j)) - gamma*alpha
    a <- sample(a_grid, 1, prob = exp(log_pi_a - max(log_pi_a)))

    # save past burn-in
    if(s > burn_in){
      SAMPLES <- rbind(SAMPLES, c(a, b, theta_j))
    }
  }
  colnames(SAMPLES) <- c("a", "b", "theta_1", "theta_2", "theta_3",
                         "theta_4", "theta_5")

  return(SAMPLES)
}
```

Part C

Simulate data from the assumed model with $J = 5$ and the n_j's set to your preferred values, but with each set to at most 25. Also, set all parameter values as you like, but make sure they are reasonable (that is, avoid very extreme values). Implement the Gibbs sampler, present point and interval estimates of the group-specific mean reaction times.

Simulated data:

```r
J <- 5
a <- 1
b <- 2

set.seed(48923)

theta_j <- rgamma(J, a, b)
```

```r
n_j <- sample(3:25, J)

sum_y_j <- numeric(J)

for (i in 1:J) {
  sum_y_j[i] <- sum(rexp(n_j[i], theta_j[i]))
}
```

I set the hyperparameters so that the expected value of b is 2 and a represents a Geometric(0.5) distribution.

Run Gibbs sampler:

```r
SAMPLES <- hier_expo_sampler(J, n_j, sum_y_j,
                             alpha = 2, beta = 1, gamma = -log(0.5))
```

Part D

Compare results from hierarchical specification to the true parameter values that you set. How well does your Gibbs sampler perform?

Recall values set:

```r
c(a, b)
```

```
## [1] 1 2
```

```r
data.frame(theta_j, n_j)
```

```
##      theta_j n_j
## 1 0.39150943  20
## 2 0.07163836  13
## 3 0.27836321  25
## 4 0.03681840   5
## 5 0.68626826  16
```

```r
SAMPLES.mcmc <- mcmc(SAMPLES, start = 1)
summary(SAMPLES.mcmc)
```

```
##
## Iterations = 1:10000
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 10000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##           Mean      SD  Naive SE Time-series SE
## a       1.1623 0.39061 0.0039061      0.0057431
## b       3.1131 1.38647 0.0138647      0.0201794
## theta_1 0.4307 0.09414 0.0009414      0.0009613
```

```
## theta_2 0.1205 0.03183 0.0003183      0.0003183
## theta_3 0.3247 0.06408 0.0006408      0.0006408
## theta_4 0.0326 0.01332 0.0001332      0.0001332
## theta_5 0.6082 0.15112 0.0015112      0.0015420
##
## 2. Quantiles for each variable:
##
##            2.5%     25%     50%     75%   97.5%
## a       1.00000 1.00000 1.00000 1.00000 2.00000
## b       1.14057 2.11758 2.85105 3.85593 6.56087
## theta_1 0.26661 0.36407 0.42449 0.49051 0.62930
## theta_2 0.06645 0.09811 0.11800 0.14029 0.18964
## theta_3 0.21240 0.27892 0.32023 0.36472 0.46027
## theta_4 0.01183 0.02277 0.03079 0.04063 0.06312
## theta_5 0.35162 0.50015 0.59446 0.70214 0.94257
```

The Gibbs sampler performs decently. All of the parameter values I set are contained within the confidence intervals produced by the Gibbs sampler. The worst estimate is that of b, where the posterior mean is 3.11 while the parameter I set was 2.