

JC STAT 638 HW 6

Jack Cunningham

6.1

Reading Data:

```
bach <- read.csv("bach.csv")
no_bach <- read.csv("nobach.csv")
```

a)

Since θ_A and θ_B share the θ term they are dependent. A joint prior distribution is justified in cases where we cannot assume independence between parameters, such as here.

b)

First we obtain the joint distribution:

$$p(\theta, y_a, y_b, \gamma) = p(y_a|\theta_a)p(y_b|\theta_b)p(\gamma)p(\theta)$$

Where:

$$p(y_a|\theta_a) = \theta_a^{y_a} e^{-n_a \theta_a} = \theta^{y_a} e^{-n_a \theta}$$

$$p(y_b|\theta_b) = \theta_b^{y_b} e^{-n_b \theta_b} = \gamma^{y_b} \theta^{y_b} e^{-n_b (\theta \gamma)}$$

$$p(\gamma) = \gamma^{a_\gamma - 1} e^{-(b_\gamma) \gamma}$$

$$p(\theta) = \theta^{a_\theta - 1} e^{-(b_\theta) \theta}$$

Combined we have:

$$p(\theta, y_a, y_b, \gamma) = \theta^{y_a + y_b + a_\theta - 1} \gamma^{a_\gamma - 1} e^{-n_a \theta - n_b \theta - b_\gamma(\gamma) - b_\theta \theta}$$

Then to obtain the form of the full conditional distribution of θ given y_a, y_b, γ we treat all non-theta terms as constants and find the proportional density:

$$p(\theta|y_a, y_b, \gamma) \propto \theta^{y_a + y_b + a_\theta - 1} e^{-\theta(n_a + \gamma n_b + b_\theta)}$$

Clearly the proportional density is $\text{gamma}(y_a + y_b + a_\theta, n_a + \gamma n_b + b_\theta)$

c)

We take the joint distribution and treat all non-gamma terms as constants to get the full conditional distribution of γ given y_a, y_b, θ :

$$p(\gamma|y_a, y_b, \theta) \propto \gamma^{a_\gamma + y_b - 1} e^{-\gamma(n_b \theta + b_\gamma)}$$

Clearly the proportional density is $\text{gamma}(y_b + a_\gamma, n_b \theta + b_\gamma)$

d)

```
set.seed(10)
#priors
a_theta = 2
b_theta = 1
a_gamma = b_gamma = c(8,16,32,64,128)
#data
y_a = sum(bach)
n_a = length(bach)
y_b = sum(no_bach)
n_b = length(no_bach)

#gibbs sampling
n_iter = 5000

theta <- numeric(n_iter)
gamma <- numeric(n_iter)
theta_diff <- numeric(5)

for (i in 1:length(a_gamma)){
  gamma[1] <- 1
  for(j in 2:n_iter){
    theta[j] <- rgamma(1, shape = y_a + y_b + a_theta,
```

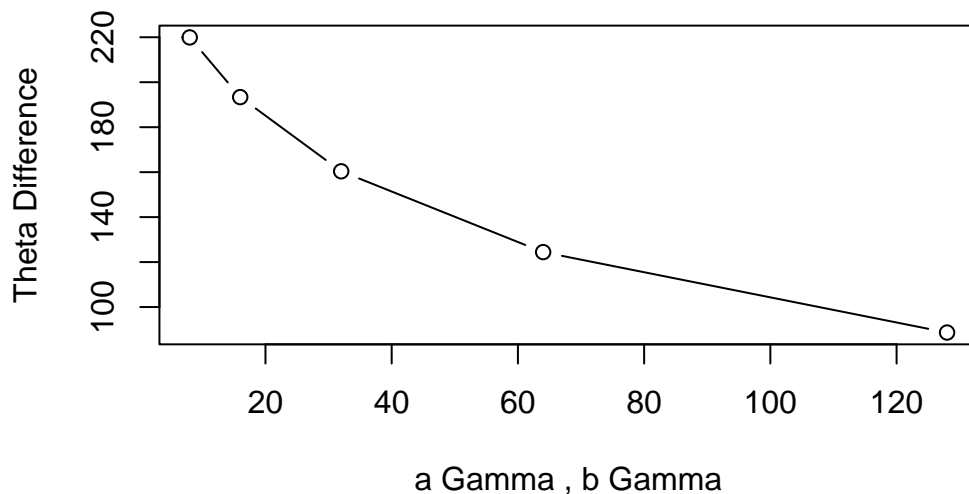
```

        rate = n_a + n_b*gamma[j - 1] + b_theta)
gamma[j] <- rgamma(1, shape = y_b + a_gamma[i],
                  rate = n_b*theta[j] + b_gamma[i])
}
theta_a <- theta
theta_b <- theta * gamma
theta_diff[i] <- mean(theta_b - theta_a)
}

plot(x = a_gamma, y = theta_diff, type = "b",
     main = "Theta B - Theta A over different priors for gamma",
     xlab = "a Gamma , b Gamma",
     ylab = "Theta Difference")

```

Theta B – Theta A over different priors for gamma



As we make our prior on γ stronger we can see that the difference we see between θ_B and θ_A tighten. This is because this prior is that of a relative rate θ_b/θ_a equal to one, or in other words an assumption that the count of children has the same distribution regardless of whether the father holds a bachelor's degree.

6.3)

```
raw_divorce_data <- read.table("divorce.dat")
df <- data.frame(age_diff = raw_divorce_data$V1, divorce = raw_divorce_data$V2)
head(df)
```

	age_diff	divorce
1	2	0
2	0	0
3	0	0
4	1	0
5	0	0
6	-1	0

a)

We only need information about z, x to have all the information we need for $p(\beta|y, x, z, c)$ so we use the Bayes rule.

$$p(\beta|y, x, z, c) = p(\beta|z, x) = p(z|\beta, x)p(\beta)$$

We know that $\beta \sim \text{normal}(0, \tau_B^2)$, so:

$$p(\beta) \propto e^{-\beta^2/2\tau_b^2}$$

We have:

$$Z_i = \beta x_i + \epsilon_i$$

We also know that $\epsilon_i \sim \text{i.i.d normal}(0,1)$, so:

$$Z_i \sim N(\beta x_i, 1)$$

Then:

$$p(z_i|\beta, x_i) \propto \exp[-\frac{1}{2}(z_i - \beta x_i)^2/1]$$

Then over all z_i we have:

$$p(z|\beta, x) \propto \exp[-\frac{1}{2} \sum_{i=1}^n (z_i - \beta x_i)^2]$$

So we end up with:

$$p(\beta|y, x, z, c) \propto \exp[-\frac{1}{2} \sum_{i=1}^n (z_i - \beta x_i)^2 + \frac{\beta^2}{\tau_b^2}]$$

After some more algebra we get to:

$$p(\beta|y, x, z, c) \propto \exp[-\frac{1}{2}(\beta^2(\sum x_i^2 + \frac{1}{\tau_b^2}) - 2\beta \sum x_i z_i + \sum z_i^2)]$$

This reduces to:

$$p(\beta|z, x) \propto \exp[-\frac{1}{2V_\beta}(\beta - \mu_\beta)^2]$$

where $V_\beta = \frac{1}{\sum_i x_i^2 + \frac{1}{\tau_b^2}}$, $\mu_\beta = V_\beta \sum_i x_i z_i$.

So:

$$\beta|x, y, z, c \sim N(\mu_B, V_B)$$

b)

We have:

$$p(c|y, x, z, \beta) = p(y|x, z, \beta, c)p(c)$$

Since we are assuming $c \sim N(0, \tau_c^2)$ we have:

$$p(c) \propto \exp[-c^2/2\tau_c^2]$$

Then we have to find the form of $p(y|x, z, \beta, c)$.

We have $Y_i = \delta_{(c, \infty)}(Z_i)$, which is an indicator function where if $Z_i > c$ we have $Y_i = 1$ and if $Z_i \leq c$ we have $Y_i = 0$. So:

$$Y_i = 1\{Z_i > c\}$$

Then, since we know each Z_i , we have some information about what c is.

For instance, if a $Y_i = 1$ we know that $Z_i > c$ or equivalently $c < Z_i$.

Similarly, if a $Y_i = 0$ we know that $Z_i < c$ or equivalently $c > Z_i$.

Then, for all Z_i we know that:

$$c_{\min} = \max[Z_i | Y_i = 0], \quad c_{\max} = \min[Z_i | Y_i = 1]$$

Then that leaves us with:

$$p(c|y, x, z, \beta) = \exp[-c^2/2\tau_c^2] \times 1\{c_{\min} < c < c_{\max}\}$$

Which we can see is proportional to the normal density, but c is constrained in the interval (c_{\min}, c_{\max}) .

The same idea carries for $p(z_i|y, x, z_{-i}, \beta, c)$:

If we know about y_i we have information that z_i is above or below c . So:

$$p(z_i|y, x, z_{-i}, \beta, c) \propto \begin{cases} N(\beta x_i, 1) & z_i > c & y_i = 1 \\ N(\beta x_i, 1) & z_i \leq c & y_i = 0 \end{cases}$$

c)

```
set.seed(2)
n <- dim(df)[1]
x <- df$age_diff
y <- df$divorce

#Priors
tau2_beta <- 16
tau2_c <- 16
tau_beta <- sqrt(tau2_beta)
tau_c <- sqrt(tau2_c)

#Initialization
beta <- 0
c <- 0
z <- numeric(n)
n_iter <- 20000
beta_store <- numeric(n_iter)
c_store <- numeric(n_iter)
```

```

# Gibbs Sample
for (iter in 1:n_iter) {
  # Z sampling
  mu <- beta * x
  for (i in 1:n) {
    if (y[i] == 1) {
      a <- (c - mu[i]) / 1
      u <- runif(1, pnorm(a), 1)
      z[i] <- mu[i] + qnorm(u)
    } else {
      b <- (c - mu[i]) / 1
      u <- runif(1, 0, pnorm(b))
      z[i] <- mu[i] + qnorm(u)
    }
  }
}

# Beta sampling
V_beta <- 1 / (sum(x^2) + 1 / tau2_beta)
mu_beta <- V_beta * sum(x * z)
beta <- rnorm(1, mu_beta, sqrt(V_beta))

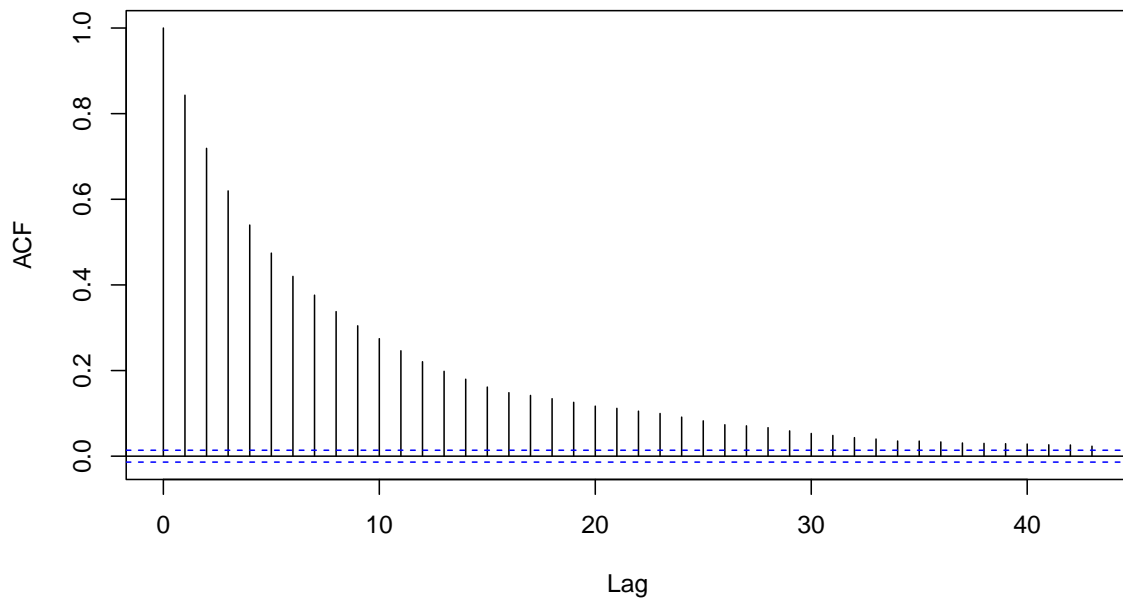
# C sampling
a <- if (any(y == 0)) max(z[y == 0]) else -Inf
b <- if (any(y == 1)) min(z[y == 1]) else Inf
lower <- pnorm((a - 0) / tau_c)
upper <- pnorm((b - 0) / tau_c)
u <- runif(1, lower, upper)
c <- 0 + tau_c * qnorm(u)

# storing
beta_store[iter] <- beta
c_store[iter] <- c
}

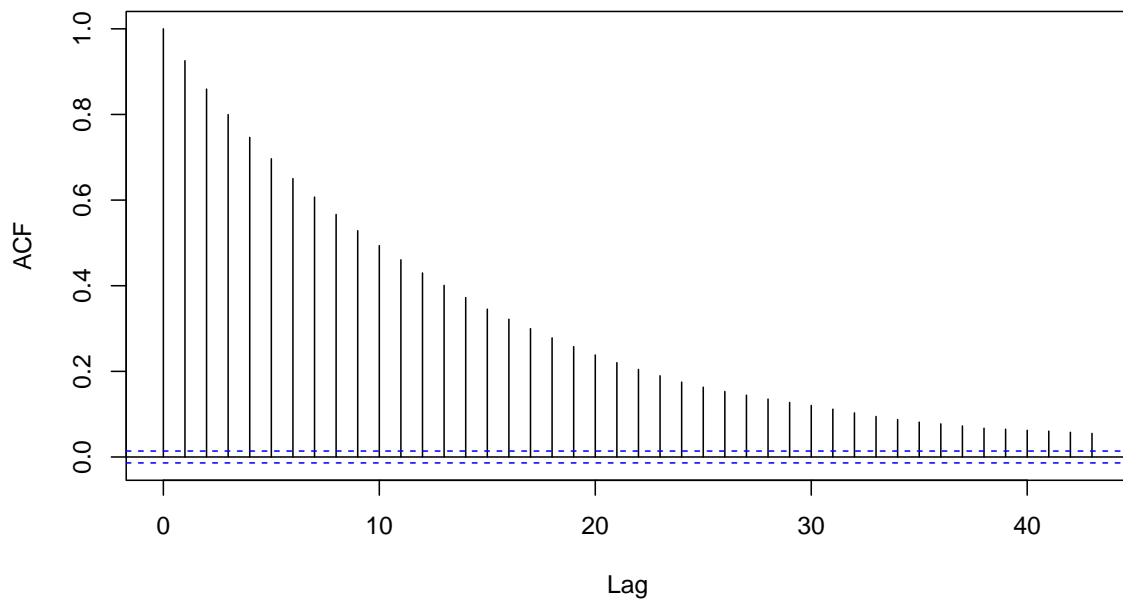
par(mfrow = c(2,1))
acf(beta_store, main = "ACF of beta")
acf(c_store, main = "ACF of c")

```

ACF of beta



ACF of c



We can see that both β and c seem to have poor mixing. The autocorrelation between samples close to each other is high and it takes about 40 lags for the autocorrelation to drop near zero. This would indicate that we may need to use more iterations to get good precision.

d)

```
cat("Posterior mean of beta:", mean(beta_store), "\n")
```

Posterior mean of beta: 0.3501144

```
cat("95% credible interval for beta:", quantile(beta_store, c(0.025, 0.975)), "\n")
```

95% credible interval for beta: 0.1035575 0.6443167

```
cat("Pr(beta > 0 | data):", mean(beta_store > 0), "\n")
```

Pr(beta > 0 | data): 0.9991