

STA 601/360 Homework 6

Yifei Wang

17 October, 2019

1. Hoff problem 6.1 Poisson population comparisons

I first derive the joint distribution of all the parameters,

$$p(\theta, \gamma, \mathbf{y}_A, \mathbf{y}_B) = p(\theta) p(\gamma) p(\mathbf{y}_A | \theta) p(\mathbf{y}_B | \theta, \gamma)$$

(a) Are θ_A and θ_B independent or dependent under this prior distribution? In what situations is such a joint prior distribution justified?

We have

$$\begin{aligned}\mathbb{E}[\theta_A] &= \mathbb{E}[\theta] = \frac{a_\theta}{b_\theta} \\ \mathbb{E}[\theta_B] &= \mathbb{E}[\theta\gamma] = \mathbb{E}[\theta]\mathbb{E}[\gamma] = \frac{a_\theta}{b_\theta} \frac{a_\gamma}{b_\gamma} \\ \mathbb{E}[\theta_A\theta_B] &= \mathbb{E}[\theta^2\gamma] = \mathbb{E}[\theta^2]\mathbb{E}[\gamma] = \frac{a_\theta + a_\theta^2}{b_\theta^2} \frac{a_\gamma}{b_\gamma} \neq \mathbb{E}[\theta_A]\mathbb{E}[\theta_B]\end{aligned}$$

Thus, θ_A and θ_B are dependent under this prior distribution. This joint prior distribution justified only when the true θ_A is proportional to θ_B .

(b) Obtain the form of the full conditional distribution of θ given \mathbf{y}_A , \mathbf{y}_B and γ .

$$\begin{aligned}p(\theta | \mathbf{y}_A, \mathbf{y}_B, \gamma) &\propto p(\mathbf{y}_A | \mathbf{y}_B, \theta, \gamma) p(\theta | \mathbf{y}_B, \gamma) \\ &\propto p(\mathbf{y}_A | \mathbf{y}_B, \theta, \gamma) p(\mathbf{y}_B | \theta, \gamma) p(\theta | \gamma) \\ &= p(\mathbf{y}_A | \theta) p(\mathbf{y}_B | \theta, \gamma) p(\theta) \\ &= p(\theta) \prod_{i=1}^{n_A} p(y_i | \theta) \prod_{j=1}^{n_B} p(y_j | \theta, \gamma) \\ &\propto \theta^{a_\theta - 1} e^{-\beta_\theta \theta} \prod_{i=1}^{n_A} \theta^{y_i} e^{-\theta} \prod_{j=1}^{n_B} \theta^{y_j} e^{-\gamma \theta} \\ &= \theta^{\sum_{i=1}^{n_A} y_i + \sum_{j=1}^{n_B} y_j + a_\theta - 1} e^{-(b_\theta + n_A + n_B \gamma) \theta}\end{aligned}$$

Using the kernel trick, we know that

$$\theta | \mathbf{y}_A, \mathbf{y}_B, \gamma \sim \text{Gamma}\left(\sum_{i=1}^{n_A} y_i + \sum_{j=1}^{n_B} y_j + a_\theta, b_\theta + n_A + n_B \gamma\right)$$

(c) Obtain the form of the full conditional distribution of γ given y_A , y_B and θ .

$$\begin{aligned}
 p(\gamma | \mathbf{y}_A, \mathbf{y}_B, \theta) &\propto p(\mathbf{y}_B | \mathbf{y}_A, \theta, \gamma) p(\gamma | \mathbf{y}_A, \theta) \\
 &\propto p(\mathbf{y}_B | \mathbf{y}_A, \theta, \gamma) p(\mathbf{y}_A | \gamma, \theta) p(\gamma | \theta) \\
 &= p(\mathbf{y}_B | \theta, \gamma) p(\mathbf{y}_A | \theta) p(\gamma) \\
 &\propto \gamma^{a_\gamma - 1} e^{-b_\gamma \gamma} \prod_{i=1}^{n_B} \gamma^{y_i} e^{-\theta \gamma} \\
 &= \gamma^{\sum_{i=1}^{n_B} y_i + a_\gamma - 1} e^{-(n\theta + b_\gamma) \gamma}
 \end{aligned}$$

Using the kernel trick, we know that,

$$\gamma | \mathbf{y}_A, \mathbf{y}_B, \theta \sim \text{Gamma}\left(\sum_{i=1}^{n_B} y_i + a_\gamma, n_B \theta + b_\gamma\right)$$

(d) Set $a_\theta = 2$ and $b_\theta = 1$. Let $a_\gamma = b_\gamma \in \{8, 16, 32, 64, 128\}$. For each of these five values, run a Gibbs sampler of at least 5,000 iterations and obtain $\mathbb{E}[\theta_B - \theta_A | \mathbf{y}_A, \mathbf{y}_B]$. Describe the effects of the prior distribution for γ on the results.

```

# data and prior
men_A = scan("menchild30bach.dat")
men_B = scan("menchild30nobach.dat")
sum_A = sum(men_A)
sum_B = sum(men_B)
n_A = length(men_A)
n_B = length(men_B)

a_theta = 2
b_theta = 1
ab_gamma = c(8, 16, 32, 64, 128)
m = length(ab_gamma)
S = 5000

# starting point
theta_0 = a_theta / b_theta
gamma_0 = 1
gibbs_theta = matrix(nrow = m, ncol = S + 1)
gibbs_gamma = matrix(nrow = m, ncol = S + 1)
gibbs_theta[,1] = theta_0
gibbs_gamma[,1] = gamma_0

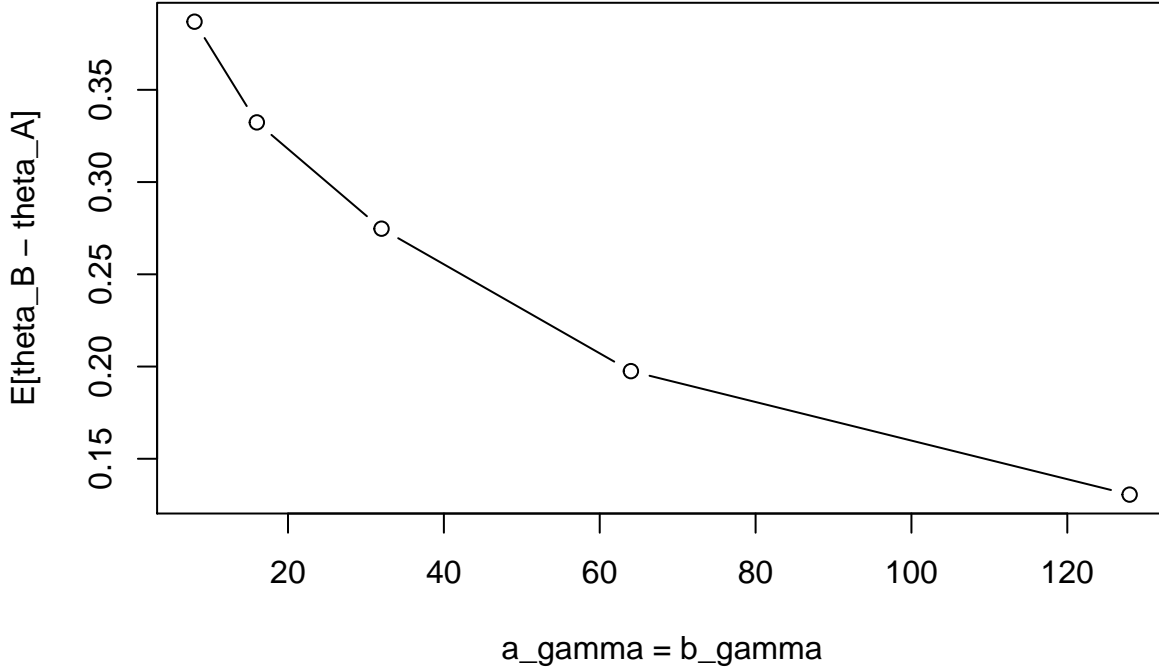
# gibbs sampler
for (s in 2:(S+1)) {
  gibbs_theta[,s] = rgamma(m, shape = sum_A + sum_B + a_theta,
                           rate = n_A + n_B * gibbs_gamma[,s-1] + b_theta)
  gibbs_gamma[,s] = rgamma(m, shape = sum_B + ab_gamma,
                           rate = n_B * gibbs_theta[,s] + ab_gamma)
}

# analysis
theta_A = gibbs_theta
theta_B = gibbs_theta * gibbs_gamma

```

```
E_BmA = rowMeans(theta_B - theta_A)
plot(ab_gamma, E_BmA, "b",
     xlab = "a_gamma = b_gamma",
     ylab = "E[theta_B - theta_A]",
     main = "Effects of Different Priors on the Results")
```

Effects of Different Priors on the Results



The plot shows that the expectation of the difference between θ_B and θ_A shrinkages to 0 when a_γ and b_γ increases towards infinity. This could be seen from the posterior mean of γ . We know that $\frac{\theta_B}{\theta_A} = \gamma$ and the posterior mean of γ is

$$\frac{\sum_{i=1}^{n_B} y_i + a_\gamma}{n_B \theta + b_\gamma} = \frac{n_B}{n_B \theta + b_\gamma} \frac{\sum_{i=1}^{n_B} y_i}{n_B} + \frac{b_\gamma}{n_B \theta + b_\gamma} \frac{a_\gamma}{b_\gamma}$$

and this converges to 1 when prior sample sizes $a_\gamma = b_\gamma \rightarrow \infty$. This makes $\mathbb{E}(\theta_B | \mathbf{y}_A, \mathbf{y}_B) = \mathbb{E}(\theta_A | \mathbf{y}_A, \mathbf{y}_B)$ and thus $\mathbb{E}(\theta_B - \theta_A | \mathbf{y}_A, \mathbf{y}_B) = 0$.

2. Hoff problem 6.3

To help tackle the problem, I first derive and factorize the joint distribution for all parameters.

$$p(x, y, z, c, \beta, \tau_\beta^2, \tau_c^2) = p(y|z, c) p(z|\beta, x) p(\beta|\tau_\beta^2) p(c|\tau_c^2) p(x)$$

(a) Assuming $\beta \sim \text{Normal}(0, \tau_\beta^2)$ obtain the full conditional distribution $p(\beta | \mathbf{y}, \mathbf{x}, \mathbf{z}, c)$.

$$\begin{aligned}
p(\beta|\mathbf{y}, \mathbf{x}, \mathbf{z}, c) &\propto p(\mathbf{z}|\mathbf{x}, \mathbf{y}, \beta, c) p(\beta|\mathbf{y}, \mathbf{x}, c) \\
&= p(\mathbf{z}|\mathbf{x}, \beta) p(\beta) \\
&\propto \prod_{i=1}^{25} \exp\left(-\frac{1}{2}(z_i - \beta x_i)^2\right) \exp\left(-\frac{1}{2\tau_\beta^2}\beta^2\right) \\
&= \exp\left(-\frac{1}{2}\left[\left(\frac{1}{\tau_\beta^2} + \sum_{i=1}^{25} x_i^2\right)\beta^2 - 2\beta \sum_{i=1}^{25} x_i z_i\right]\right) \\
&= \exp\left(-\frac{1}{2}\left[\left(\frac{1}{\tau_\beta^2} + \mathbf{x}^T \mathbf{x}\right)\beta^2 - 2\beta \mathbf{x}^T \mathbf{z}\right]\right) \\
&= \exp\left(-\frac{1}{2}\left(\frac{1}{\tau_\beta^2} + \mathbf{x}^T \mathbf{x}\right)\left(\beta - \left(\frac{1}{\tau_\beta^2} + \mathbf{x}^T \mathbf{x}\right)^{-1} \mathbf{x}^T \mathbf{z}\right)^2\right) \\
&\sim \text{Normal}\left(\left(\frac{1}{\tau_\beta^2} + \mathbf{x}^T \mathbf{x}\right)^{-1} \mathbf{x}^T \mathbf{z}, \left(\frac{1}{\tau_\beta^2} + \mathbf{x}^T \mathbf{x}\right)^{-1}\right)
\end{aligned}$$

(b) Assuming $c \sim \text{Normal}(0, \tau_c^2)$, show that $p(c|\mathbf{y}, \mathbf{x}, \mathbf{z}, \beta)$ is a constrained normal density, i.e. proportional to a normal density but constrained to lie in an interval. Similarly, show that $p(z_i|\mathbf{y}, \mathbf{x}, \mathbf{z} - i, \beta, c)$ is proportional to a normal density but constrained to be either above c or below c , depending on y_i .

Let Truncated Normal(μ, σ^2, a, b) be the truncated Normal distribution of Normal(μ, σ^2) on $[a, b]$. For the full conditional distribution of c , we have

$$\begin{aligned}
p(c|\mathbf{y}, \mathbf{x}, \mathbf{z}, \beta) &\propto p(\mathbf{y}|\mathbf{x}, \mathbf{z}, \beta, c) p(c|\mathbf{x}, \mathbf{z}, \beta) \\
&= p(\mathbf{y}|\mathbf{z}, c) p(c) \\
&= \prod_{i=1}^{25} p(y_i|z_i, c) p(c) \\
&= \prod_{i=1}^{25} \sigma_{c, \infty}(z_i) p(c) \\
&= [\max_{y_i=0}(z_i), \min_{y_i=1}(z_i)] p(c)
\end{aligned}$$

so we have,

$$c|\mathbf{y}, \mathbf{x}, \mathbf{z}, \beta \sim \text{Truncated Normal}(0, \tau_c^2, \max_{y_i=0}(z_i), \min_{y_i=1}(z_i))$$

We know that $Z_i = \beta x_i + \epsilon_i$ and $\epsilon_i \sim \text{Normal}(0, 1)$ so we have $Z_i \sim \text{Normal}(\beta x_i, 1)$. As for the full conditional distribution of \mathbf{z} , we know any z_i is independent on z_j for any $j \neq i$. Thus we have

$$\begin{aligned}
p(z_i|\mathbf{y}, \mathbf{x}, \mathbf{z}_{-i}, \beta, c) &= p(z_i|\mathbf{y}, \mathbf{x}, \beta, c) \\
&= p(z_i|y_i, x_i, \beta, c) \\
&\propto p(y_i|z_i, x_i, \beta, c) p(z_i|x_i, \beta, c) \\
&= p(y_i|z_i, c) p(z_i|x_i, \beta) \\
&= f(z_i)
\end{aligned}$$

where

$$f(z_i) = \begin{cases} p(z_i|x_i, \beta)[z_i < c], & \text{if } y_i = 0 \\ p(z_i|x_i, \beta)[z_i > c], & \text{if } y_i = 1 \end{cases}$$

and

$$z_i \mid y_i = 0 \sim \text{Truncated Normal}(\beta x_i, 1, -\infty, c)$$

$$z_i \mid y_i = 1 \sim \text{Truncated Normal}(\beta x_i, 1, c, \infty)$$

(c) Letting $\tau_\beta^2 = \tau_c^2 = 16$, implement a Gibbs sampling scheme that approximates the joint posterior distribution of Z , β , and c (a method for sampling from constrained normal distributions is outlined in Section 12.1.1). Run the Gibbs sampler long enough so that the effective sample sizes of all unknown parameters are greater than 1,000 (including the Z_i 's). Compute the autocorrelation function of the parameters and discuss the mixing of the Markov chain.

```
# data and priors
set.seed(323)
library(truncnorm)
library(coda)
xy = scan("divorce.dat")
x = xy[seq(1, 49, 2)]
y = xy[seq(2, 50, 2)]

tau_b = tau_c = 4
beta_sigma2 = 1 / (1/tau_b^2 + sum(x^2))
S = 50000

gibbs_beta = vector(mode = "numeric", length = S + 1)
gibbs_c = vector(mode = "numeric", length = S + 1)
gibbs_z = matrix(nrow = length(x), ncol = S + 1)

# full conditional distribution
rc_gibbs = function(y, z) {
  a = max(z[y==0])
  b = min(z[y==1])
  c = rtruncnorm(1, a = a, b = b, mean = 0, sd = tau_c^2)
  return(c)
}

rz_gibbs = function(x, y, beta, c) {
  z_y0 = (1 - y) * rtruncnorm(25, b = c, mean = beta*x, sd = 1)
  z_y1 = y * rtruncnorm(25, a = c, mean = beta*x, sd = 1)
  return(z_y0 + z_y1)
}

# starting point
gibbs_beta_0 = 0
gibbs_c_0 = 0
gibbs_z_0 = rz_gibbs(x = x, y = y, beta = gibbs_beta_0, c = gibbs_c_0)
gibbs_beta[1] = gibbs_beta_0
gibbs_c[1] = gibbs_c_0
gibbs_z[, 1] = gibbs_z_0

# gibbs sampler
```

```

for (s in 2:(S+1)) {
  beta_mu = beta_sigma2 * sum(x*gibbs_z[,s-1])
  gibbs_beta[s] = rnorm(1, mean = beta_mu, sd = sqrt(beta_sigma2))
  gibbs_c[s] = rc_gibbs(y = y, z = gibbs_z[,s-1])
  gibbs_z[, s] = rz_gibbs(x = x, y = y, beta = gibbs_beta[s], c = gibbs_c[s])
}

# effective sizes
es = data.frame(parameter = vector(length = 25), EffectiveSize = vector(length = 25), geq1000 = vector(
es[1,] = c("beta", floor(effectiveSize(gibbs_beta)), 0)
es[2,] = c("c", floor(effectiveSize(gibbs_c)), 0)
for (i in 1:25) {
  es[i+2,] = c(paste("z", i, sep="_"), floor(effectiveSize(gibbs_z[i,])), 0)
}
es$geq1000 = es$EffectiveSize > 1000
es

```

```

##      parameter EffectiveSize geq1000
## 1         beta         2860      TRUE
## 2           c         1836      TRUE
## 3         z_1         8790      TRUE
## 4         z_2        27761      TRUE
## 5         z_3        27094      TRUE
## 6         z_4        13472      TRUE
## 7         z_5        25579      TRUE
## 8         z_6        42664      TRUE
## 9         z_7         6230      TRUE
## 10        z_8         4026      TRUE
## 11        z_9        24875      TRUE
## 12       z_10         8734      TRUE
## 13       z_11        37592      TRUE
## 14       z_12         8249      TRUE
## 15       z_13         5399      TRUE
## 16       z_14         7859      TRUE
## 17       z_15         6668      TRUE
## 18       z_16        40762      TRUE
## 19       z_17         7252      TRUE
## 20       z_18         8672      TRUE
## 21       z_19        26773      TRUE
## 22       z_20        25726      TRUE
## 23       z_21         8055      TRUE
## 24       z_22        25511      TRUE
## 25       z_23         6430      TRUE
## 26       z_24         3392      TRUE
## 27       z_25        16931      TRUE

```

For the diagnostic of autocorrelation and the analysis of mixing of the Markov Chains, I choose parameter c , β , z_{24} and z_6 to focus on. All the z_i are independent of each other and z_{24} has the smallest effective size while z_6 has the greatest effective size. They could represent the autocorrelation and mixing characteristics of these chains. Additionally, the same methods could be easily applied to other parameters of interest.

```

# autocorrelation
autocorr.diag(as.mcmc(gibbs_beta)) # autocorrelation of different lag values

```

```

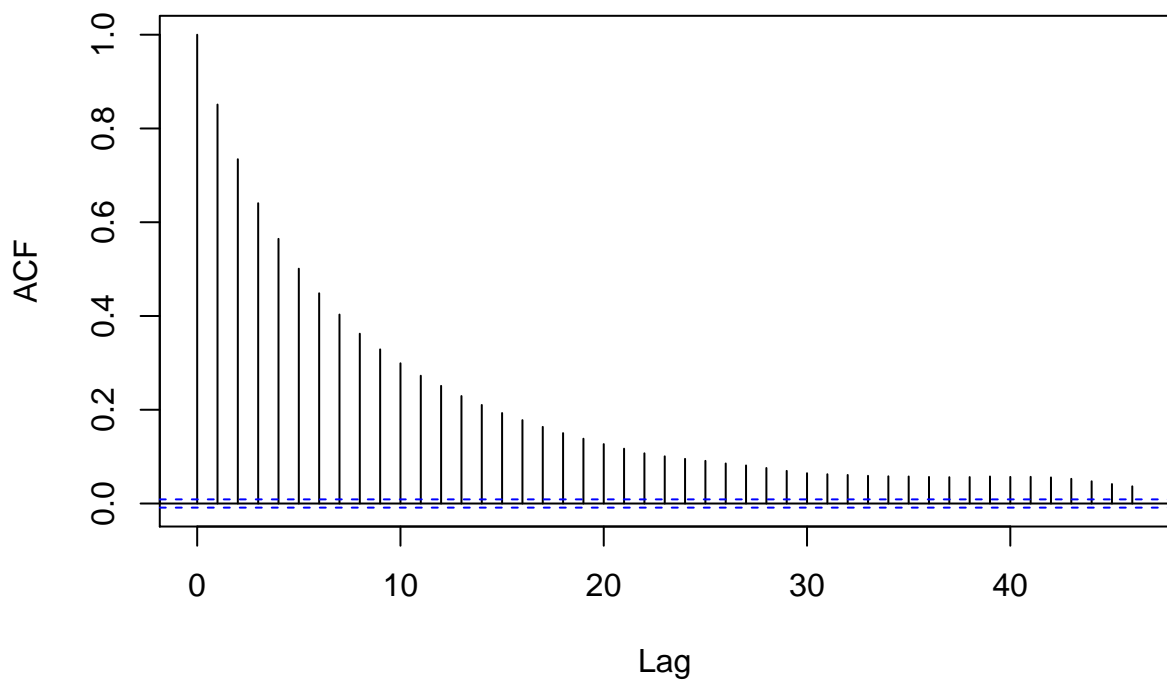
##           [,1]
## Lag 0  1.00000000

```

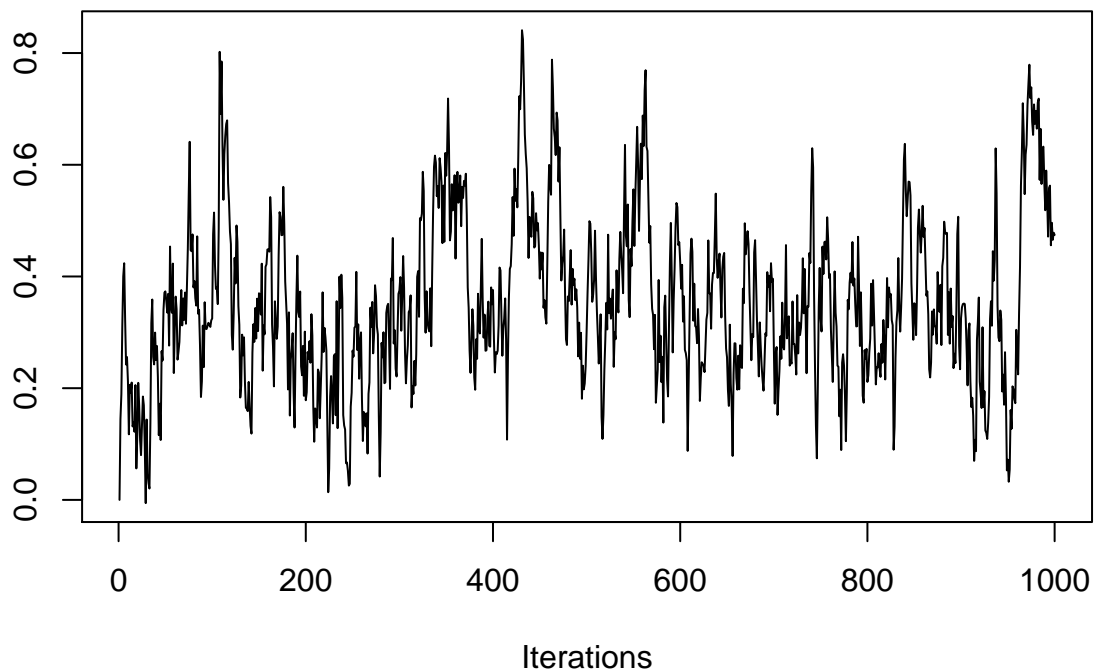
```
## Lag 1  0.85105072
## Lag 5  0.50090802
## Lag 10 0.29913686
## Lag 50 0.02358467
```

```
acf(gibbs_beta) # acf plots the decrease of autocorrelation when lag increases
```

Series gibbs_beta



```
traceplot(as.mcmc(gibbs_beta[1:1000])) # traceplots shows the mixing of mcmc
```

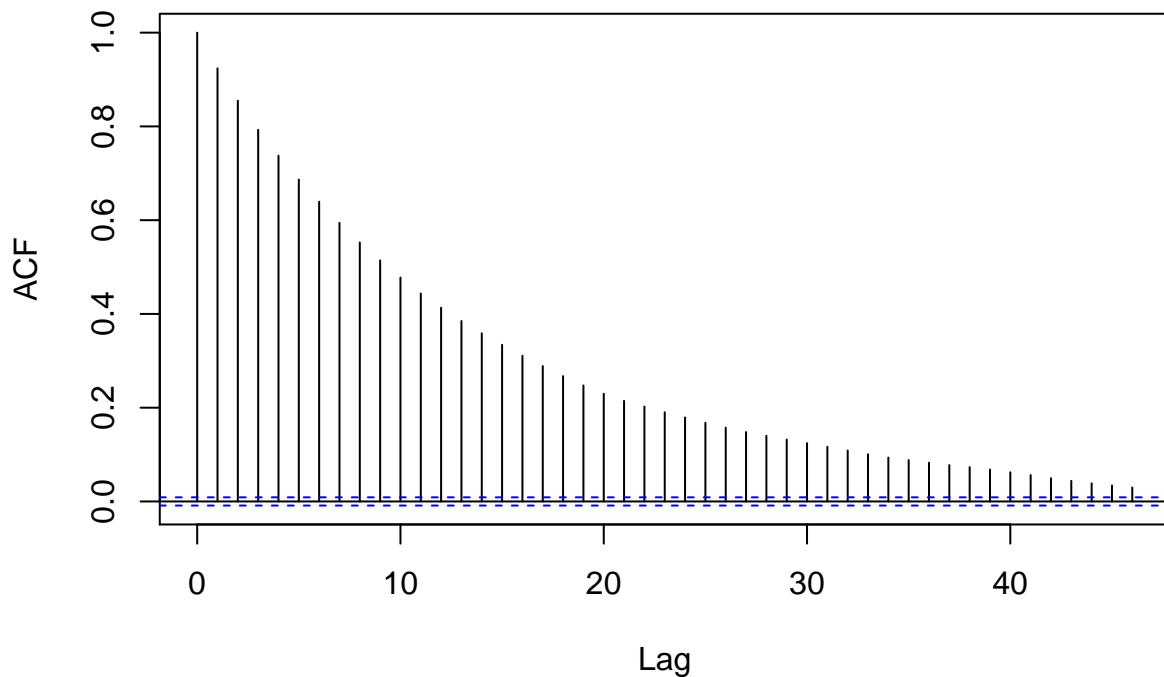


```
autocorr.diag(as.mcmc(gibbs_c)) # autocorrelation of different lag values
```

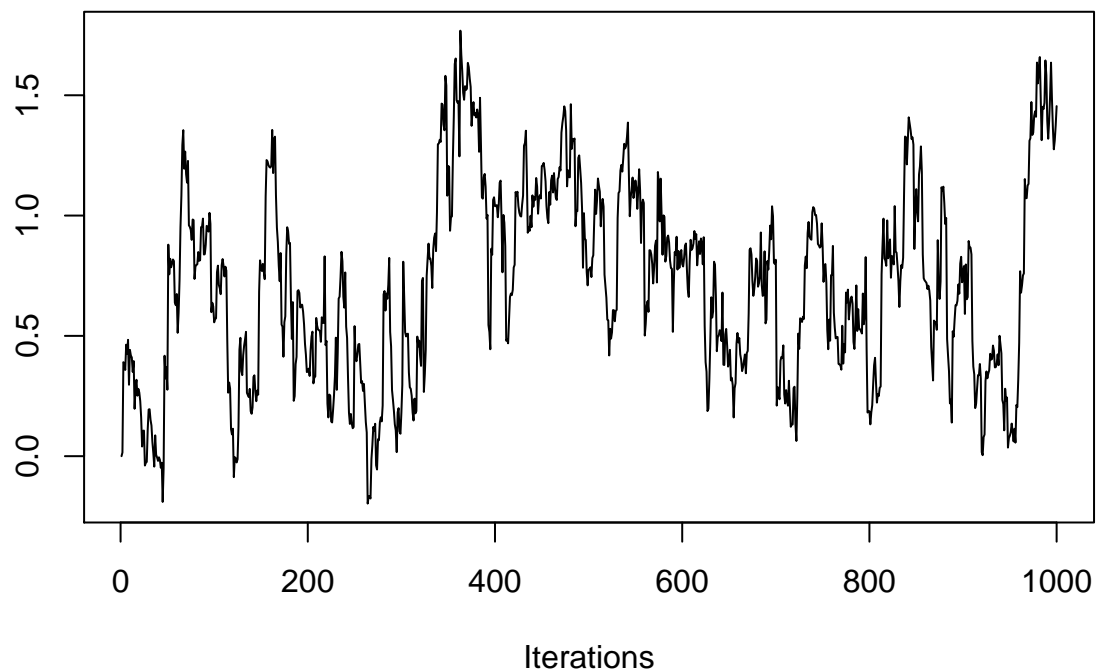
```
##           [,1]  
## Lag 0  1.00000000  
## Lag 1  0.92393182  
## Lag 5  0.68658180  
## Lag 10 0.47761489  
## Lag 50 0.01686816
```

```
acf(gibbs_c) # acf plots the decrease of autocorrelation when lag increases
```

Series gibbs_c



```
traceplot(as.mcmc(gibbs_c[1:1000])) # traceplots shows the mixing of mcmc
```

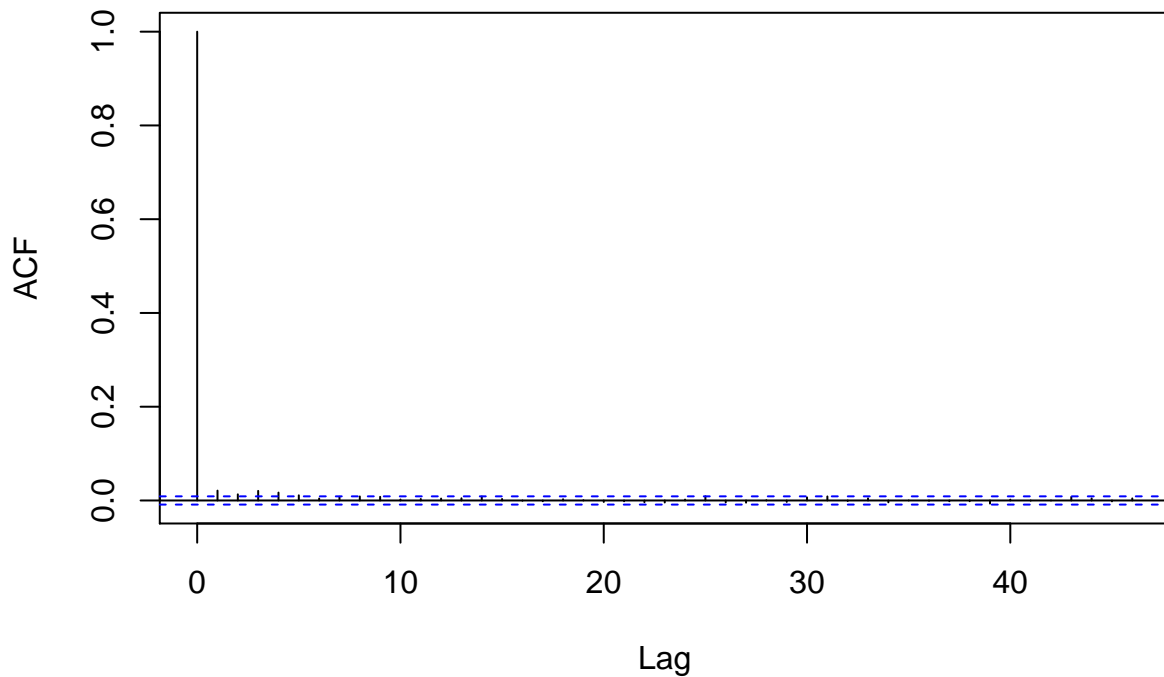



```
autocorr.diag(as.mcmc(gibbs_z[6,])) # autocorrelation of different lag values
```

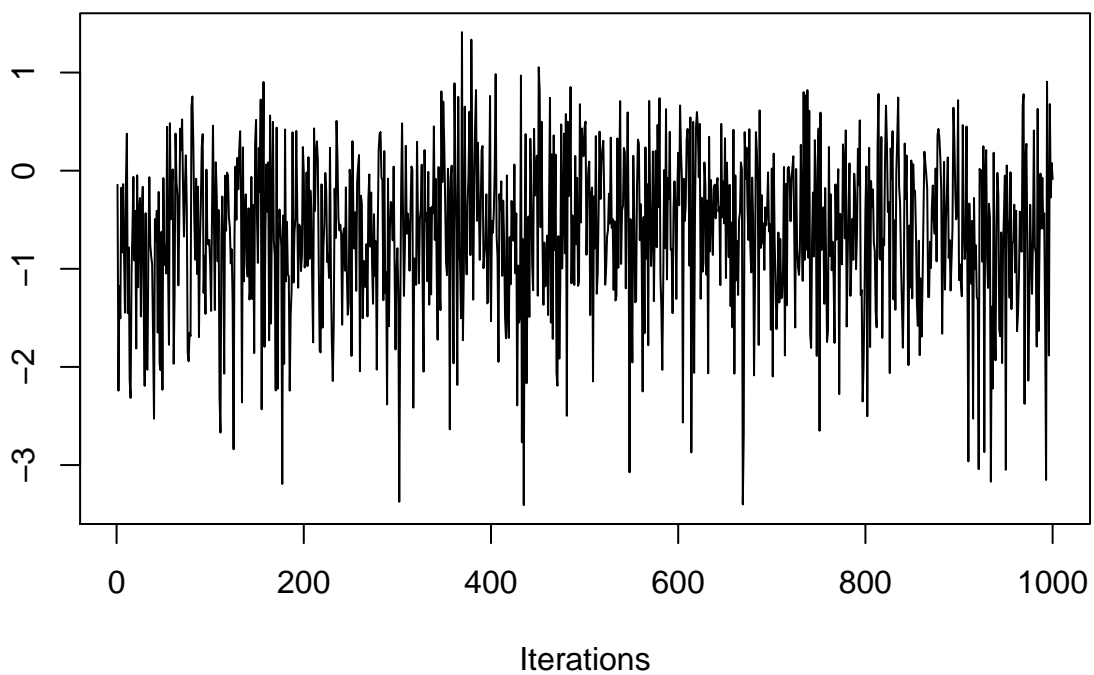
```
##           [,1]
## Lag 0  1.000000000
## Lag 1  0.021038848
## Lag 5  0.010985188
## Lag 10 0.002051234
## Lag 50 0.005900345
```

```
acf(gibbs_z[6,]) # acf plots the decrease of autocorrelation when lag increases
```

Series gibbs_z[6,]



```
traceplot(as.mcmc(gibbs_z[6,1:1000])) # traceplots shows the mixing of mcmc
```



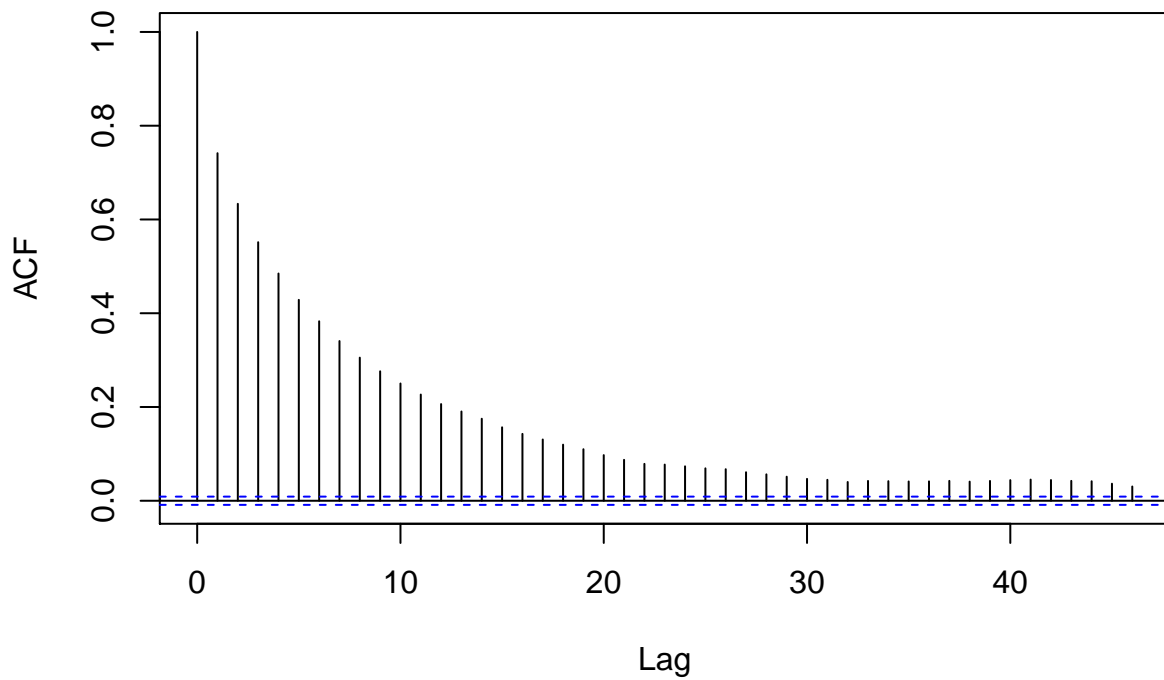
```
autocorr.diag(as.mcmc(gibbs_z[24,])) # autocorrelation of different lag values
```

```
##           [,1]
## Lag 0  1.0000000
## Lag 1  0.74142925
## Lag 5  0.42856777
```

```
## Lag 10 0.25016098  
## Lag 50 0.01547791
```

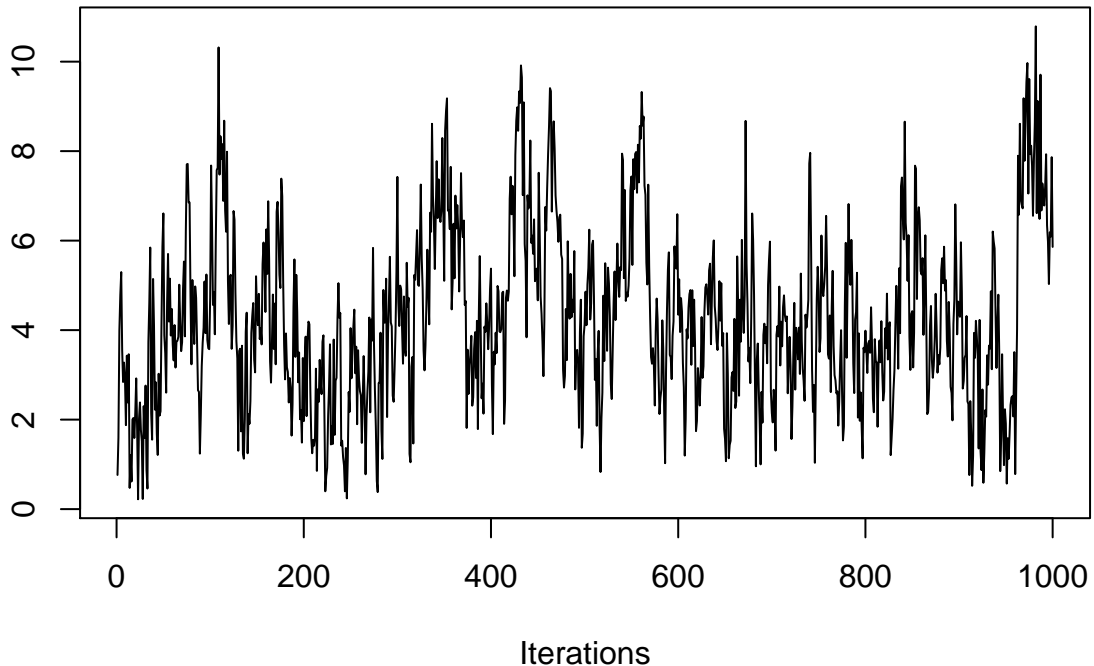
```
acf(gibbs_z[24,]) # acf plots the decrease of autocorrelation when lag increases
```

Series gibbs_z[24,]



```
traceplot(as.mcmc(gibbs_z[24,1:1000]), main = "Traceplot of z_24") # traceplots shows the mixing of mcm
```

Traceplot of z_{24}



From the tables and plots above we can tell that

1. Chains with smaller effective size often have a more severe mixing and autocorrelation problem.
2. Parameter c has the smallest effective size and the most severe mixing and autocorrelation problem. Its lag-10 autocorrelation is still 0.48. The “bouncing” ability shown in the traceplot is bad, which also indicates a high autocorrelation. The chain seems to quickly reach the stationary area of the posterior distribution.
3. Parameter β also has a small effective size and severe mixing and autocorrelation problem. Its lag-10 autocorrelation is still 0.30. The “bouncing” ability shown in the traceplot is not good, which also indicates a high autocorrelation. The chain seems to quickly reach the stationary area of the posterior distribution.
4. Parameter z_i in general has large effective sizes and the values vary among different i . It seems that the effective sizes of z_i when $y_i = 1$ is generally smaller than those of z_j when $y_j = 0$. In general, the mixing and autocorrelation problem is not that bad for all the z_i 's. The lag-10 autocorrelation is 0.002 for z_6 and 0.25 for z_{24} . The “bouncing” ability shown in the traceplot is good for z_6 and not that great for z_{24} , which also indicates a high autocorrelation for z_{24} . Both chains seem to quickly reach the stationary area of the posterior distribution.

(d) Obtain a 95% posterior confidence interval for β , as well as $\Pr(\beta > 0|y, x)$.

```
# 95% posterior confidence interval for beta
quantile(gibbs_beta, probs = c(0.025, 0.975))
```

```
##      2.5%      97.5%
## 0.1011909 0.6535880
```

```
# posterior probability of beta > 0
mean(gibbs_beta > 0)
```

```
## [1] 0.999
```

3. Hoff problem 7.3

(a) For each of the two species, obtain posterior distributions of the population mean θ and covariance matrix Σ as follows: Using the semi-conjugate prior distributions for θ and Σ , set μ_0 equal to the sample mean of the data, Λ_0 and S_0 equal to the sample covariance matrix and $\nu_0 = 4$. Obtain 10,000 posterior samples of θ and Σ . Note that this “prior” distribution loosely centers the parameters around empirical estimates based on the observed data (and is very similar to the unit information prior described in the previous exercise). It cannot be considered as our true prior distribution, as it was derived from the observed data. However, it can be roughly considered as the prior distribution of someone with weak but unbiased information.

For the semi-conjugate priors of $\text{Normal}(\theta, \Sigma)$, we have

$$\begin{aligned}\theta &\sim \text{Normal}(\mu_0, \Lambda_0) \\ \Sigma &\sim \text{inverse-Wishart}(\nu_0, S_0^{-1})\end{aligned}$$

Then we have the full conditional posterior distribution

$$\begin{aligned}\theta | \mathbf{y}_1, \dots, \mathbf{y}_n, \Sigma &\sim \text{Normal}(\mu_n, \Lambda_n) \\ \Sigma | \mathbf{y}_1, \dots, \mathbf{y}_n, \theta &\sim \text{inverse-Wishart}(\nu_0 + n, [S_0 + S_\theta]^{-1})\end{aligned}$$

where

$$\begin{aligned}\Lambda_n &= (\Lambda_0^{-1} + n\Sigma^{-1})^{-1} \\ \mu_n &= (\Lambda_0^{-1} + n\Sigma^{-1})^{-1}(\Lambda_0^{-1}\mu_0 + n\Sigma^{-1}\bar{\mathbf{y}}) \\ S_\theta &= \sum_{i=1}^n (\mathbf{y}_i - \theta)(\mathbf{y}_i - \theta)^T\end{aligned}$$

```
# data and priors
set.seed(323)
b = scan("bluecrab.dat")
o = scan("orangecrab.dat")
blue = matrix(nrow = 50, ncol = 2)
orange = matrix(nrow = 50, ncol = 2)
blue[,1] = b[seq(1, 99, 2)]
blue[,2] = b[seq(2, 100, 2)]
orange[,1] = o[seq(1, 99, 2)]
orange[,2] = o[seq(2, 100, 2)]

mu0_blue = colMeans(blue)
mu0_orange = colMeans(orange)
L0_blue = S0_blue = cov(blue)
L0_orange = S0_orange = cov(orange)
nu0 = 4
S = 10000

# gibbs sampler
crab_gibbs = function(S, Y, mu0, L0, nu0, S0) {
  n = nrow(Y)
  p = ncol(Y)
  ybar = colMeans(Y)
```

```

THETA = array(dim = c(S, p))
SIGMA = array(dim = c(S, p, p))
Sigma = cov(Y)
for (s in 1:S) {
  # theta
  Ln = solve(solve(L0) + n*solve(Sigma))
  mun = Ln %*% (solve(L0) %*% mu0 + n*solve(Sigma) %*% ybar)
  theta = MASS::mvrnorm(n = 1, mu = mun, Sigma = Ln)

  # Sigma
  Sn = S0 + (t(Y) - c(theta)) %*% t(t(Y) - c(theta))
  Sigma = solve(rWishart(1, nu0 + n, solve(Sn))[,1])

  # update
  THETA[s,] = theta
  SIGMA[s,,] = Sigma
}
return(list(theta = THETA, sigma = SIGMA))
}

```

```

gibbs_blue = crab_gibbs(S, blue, mu0_blue, L0_blue, nu0, S0_blue)
gibbs_orange = crab_gibbs(S, orange, mu0_orange, L0_orange, nu0, S0_orange)

```

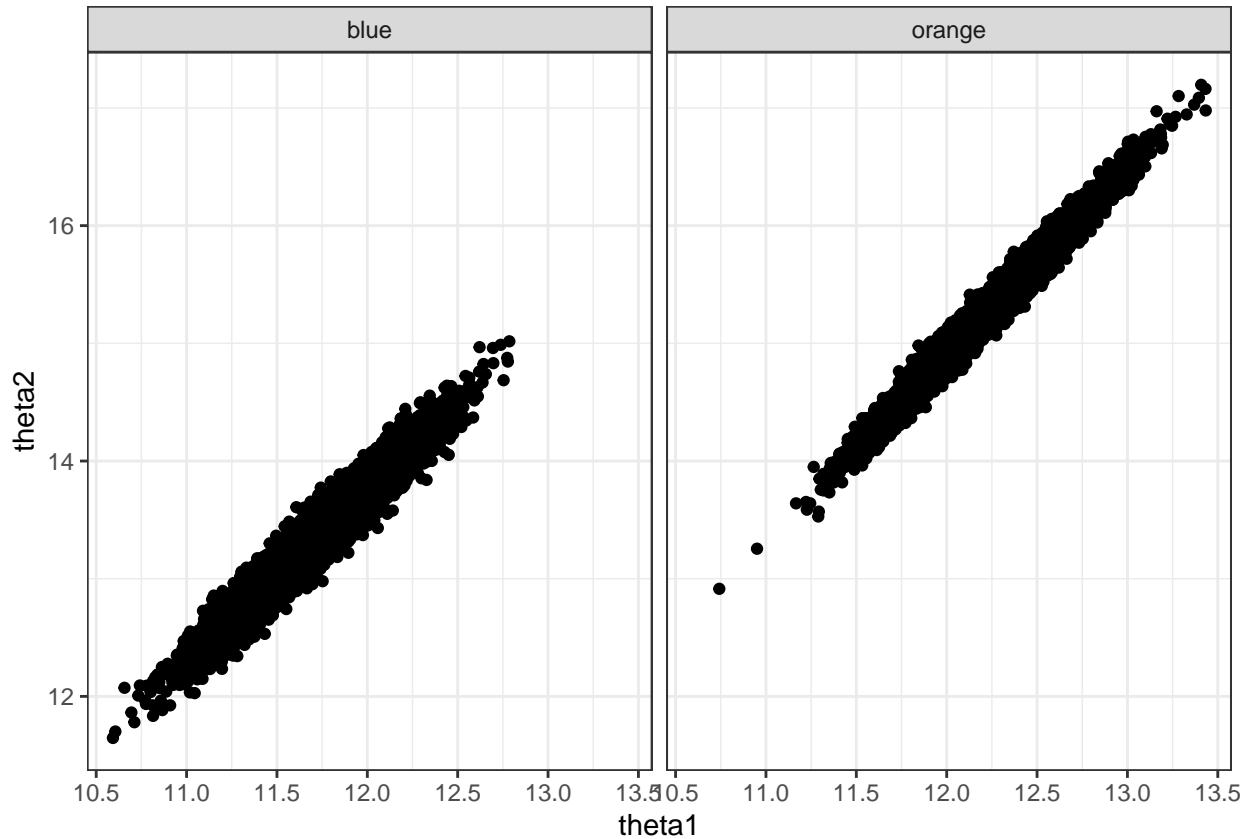
(b) Plot values of $\theta = (\theta_1, \theta_2)$ for each group and compare. Describe any size differences between the two groups.

```

theta_df_blue = data.frame(theta1 = gibbs_blue$theta[, 1], theta2 = gibbs_blue$theta[, 2], types = 'blue')
theta_df_orange = data.frame(theta1 = gibbs_orange$theta[, 1], theta2 = gibbs_orange$theta[, 2], types = 'orange')
theta_df = rbind(theta_df_blue, theta_df_orange)

ggplot(theta_df, aes(x = theta1, y = theta2)) +
  geom_point() +
  facet_wrap(~ types)

```



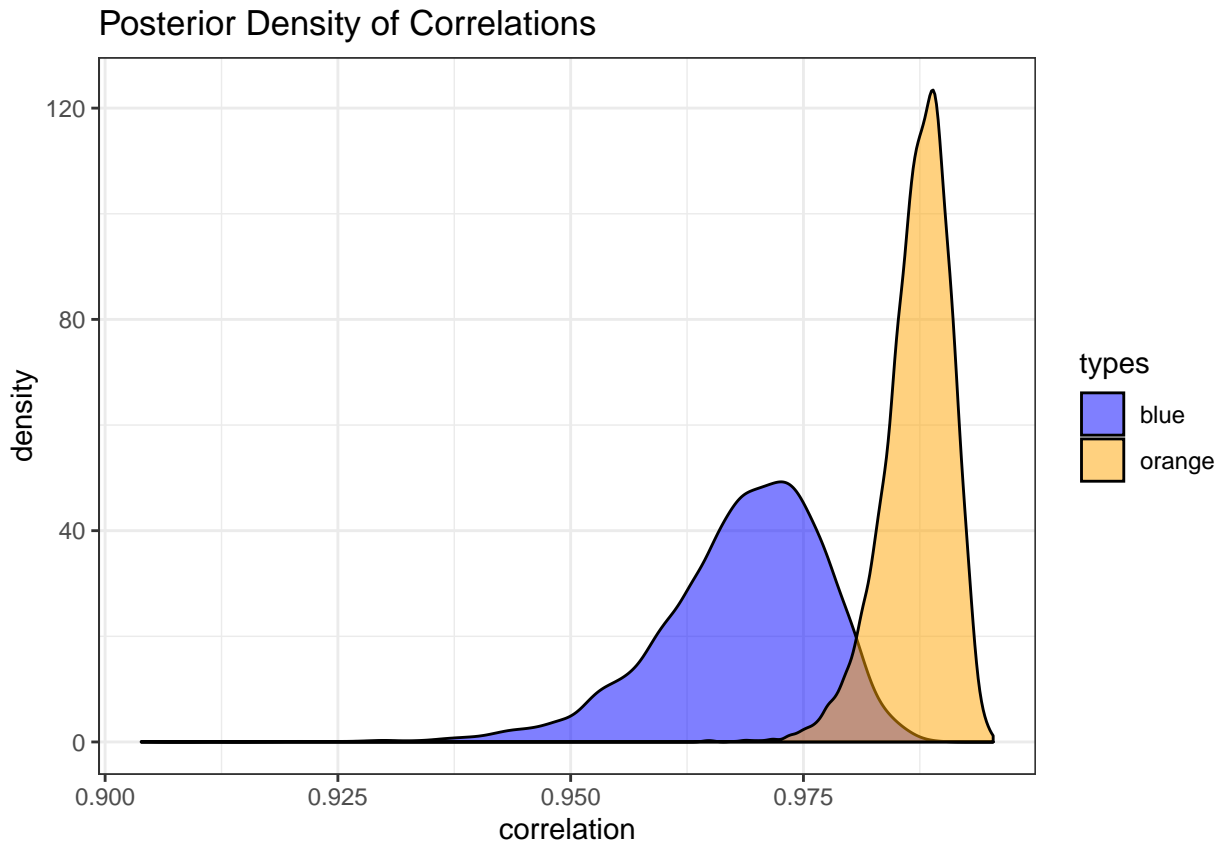
From the plot above we can tell that both θ_1 and θ_2 in blue crabs are smaller than those in orange crabs with a high probability. This means that the means of both body depth (Y_1) and rear width (Y_2) of blue crabs are smaller than those of orange crabs with a high probability. Also, there is a positive relationship between body depth and rear width for both types of crabs.

(c) From each covariance matrix obtained from the Gibbs sampler, obtain the corresponding correlation coefficient. From these values, plot posterior densities of the correlations ρ_{blue} and ρ_{orange} for the two groups. Evaluate differences between the two species by comparing these posterior distributions. In particular, obtain an approximation to $\Pr(\rho_{\text{blue}} < \rho_{\text{orange}} | y_{\text{blue}}, y_{\text{orange}})$. What do the results suggest about differences between the two populations?

```
b_sig = gibbs_blue$sigma
o_sig = gibbs_orange$sigma
corr_blue = b_sig[, 1, 2] / sqrt(b_sig[, 1, 1] * b_sig[, 2, 2])
corr_orange = o_sig[, 1, 2] / sqrt(o_sig[, 1, 1] * o_sig[, 2, 2])

corr = data.frame(correlation = c(corr_blue, corr_orange), types = rep(c("blue", "orange"), each = 5))

ggplot(corr, aes(x = correlation, fill = types)) +
  geom_density(alpha = 0.5) +
  scale_fill_manual(values = c("blue", "orange")) +
  ggtitle("Posterior Density of Correlations")
```



From the plot above we can find that the correlation of body depth and rear width in blue crabs are smaller than the correlation in orange crabs with a high probability. Also, the spread (variance) of correlation in blue crabs is larger than the one in orange crabs.

```
diff_bo = mean(corr_blue < corr_orange)
diff_bo
```

```
## [1] 0.9896
```

$\Pr(\rho_{\text{blue}} < \rho_{\text{orange}} | \mathbf{y}_{\text{blue}}, \mathbf{y}_{\text{orange}}) = 0.9896$. All these results suggest that the $\rho_{\text{blue}} < \rho_{\text{orange}}$ is true for almost for every sample. Orange crabs have a greater correlation between body depth and rear width than blue crabs.