1. Hoff 6.1

Since $\theta \sim gamma(a_\theta, b_\theta)$, $\gamma \sim gamma(a_\gamma, b_\gamma)$

θ and γ are indep

a. $Cov(\theta_A, \theta_B) = E[\theta_A \theta_B] - E[\theta_A] \cdot E[\theta_B]$

$= E[\theta^2 \gamma] - E[\theta] \cdot E[\theta\gamma]$

$= E[\theta^2] \cdot E[\gamma] - E[\theta] \cdot E[\theta] \cdot E[\gamma]$

$= E[\theta^2] \cdot E[\gamma] - (E[\theta])^2 \cdot E[\gamma]$

$= \left(E[\theta^2] - (E[\theta])^2\right) \cdot E[\gamma]$

$= Var[\theta] \cdot E[\gamma] = \frac{a_\theta}{b_\theta^2} \cdot \frac{a_\gamma}{b_\gamma}$

$\neq 0$

Since $Cov(\theta_A, \theta_B) \neq 0$, θ_A and θ_B are independent.

The prior is justified if we have reason to believe that θ_B is some product of θ_A + random Gamma-distributed noise

b. The sampling model for y_A and y_B are:

$$\begin{cases} P(y_{A1}, \ldots, y_{An} \mid \theta) \propto \theta^{\sum_{i=1}^{n} y_{Ai}} e^{-n\theta} \\ P(y_{B1}, \ldots, y_{Bn} \mid \theta, \gamma) \propto (\theta\gamma)^{\sum_{i=1}^{n} y_{Bi}} e^{-n(\theta\gamma)} \end{cases} \qquad \begin{cases} \theta \sim Ga(a_\theta, b_\theta) \\ \gamma \sim Ga(a_\gamma, b_\gamma) \end{cases}$$

Joint distribution:

$$P(y_A, y_B, \theta, \gamma \mid a_\theta, b_\theta, a_\gamma, b_\gamma) = P(y_A \mid \theta) \cdot P(y_B \mid \theta, \gamma) \cdot P(\theta \mid a_\theta, b_\theta) \cdot P(\gamma \mid a_\gamma, b_\gamma)$$

Full conditional of θ:

$$P(\theta \mid y_A, y_B, \gamma) \propto P(y_A \mid \theta) \cdot P(y_B \mid \theta, \gamma) \cdot P(\theta \mid a_\theta, b_\theta)$$

$$\propto \theta^{\sum y_{Ai}} e^{-n_A\theta} \cdot (\theta\gamma)^{\sum y_{Bi}} \cdot e^{-n_B(\theta\gamma)} \cdot \theta^{a_\theta - 1} \cdot e^{-b_\theta \theta}$$

$$\propto \theta^{\sum y_{Ai} + \sum y_{Bi} + a_\theta - 1} e^{-(n_A + n_B\gamma + b_\theta)\cdot\theta}$$

$$\therefore \theta \mid y_A, y_B, \gamma \sim Gamma\left(\sum_{i=1}^{n_A} y_{Ai} + \sum_{i=1}^{n_B} y_{Bi} + a_\theta,\ n_A + n_B\gamma + b_\theta\right)$$

c. Full conditional of γ:

$$P(\gamma \mid y_A, y_B, \theta) \propto P(y_B \mid \theta, \gamma) \cdot P(\gamma \mid a_\gamma, b_\gamma)$$

$$\propto (\theta\gamma)^{\sum y_{Bi}} e^{-n_B(\theta\gamma)} \cdot \gamma^{a_\gamma - 1} \cdot e^{-b_\gamma \cdot \gamma}$$

$$\propto \gamma^{\sum y_{Bi}} \cdot e^{-n_B(\theta\gamma)} \cdot \gamma^{a_\gamma - 1} \cdot e^{-b_\gamma \cdot \gamma}$$

$$\propto \gamma^{\sum y_{Bi} + a_\gamma - 1} e^{-(n_B\theta + b_\gamma)\gamma}$$

$$\therefore \gamma \mid y_A, y_B, \theta \sim Gamma\left(\sum_{i=1}^{n_B} y_{Bi} + a_\gamma,\ n_B\theta + b_\gamma\right)$$

see d) in r portion of homework (page 3).

Homework 4

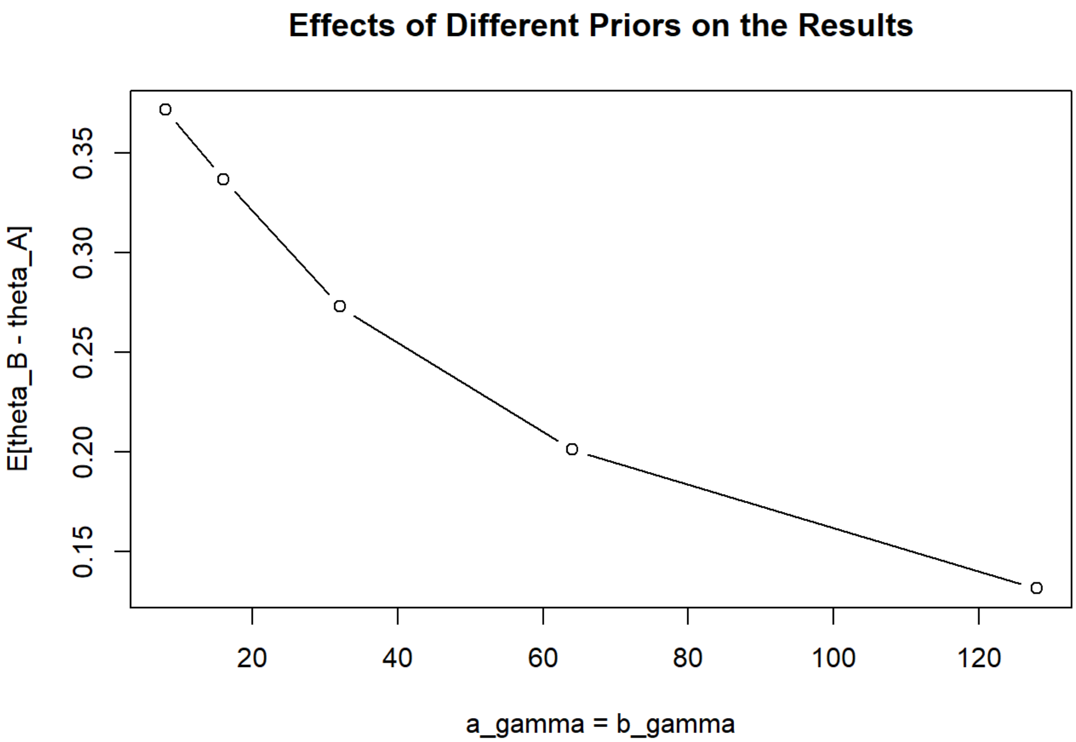
Bingying Liu

February 9, 2020

Question 1: Hoff 6.1 (d)

d) Descirbe the effects of the prior distribution for γ on the results.

```
## Q1: Hoff 6.1

# data and prior
man_A <- c(1, 0, 0, 1, 2, 2, 1, 5, 2, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 2, 1, 3,
           2, 0, 0, 3, 0, 0, 0, 2, 1, 0, 2, 1, 0, 0, 1, 3, 0, 1, 1, 0, 2, 0, 0, 2, 2, 1,
           3, 0, 0, 0, 1, 1)
man_B <- c(2, 2, 1, 1, 2, 2, 1, 2, 1, 0, 2, 1, 1, 2, 0, 2, 2, 0, 2, 1, 0, 0, 3, 6, 1, 6,
           4, 0, 3, 2, 0, 1, 0, 0, 0, 3, 0, 0, 0, 0, 0, 1, 0, 4, 2, 1, 0, 0, 1, 0, 3, 2,
           5, 0, 1, 1, 2, 1, 2, 1, 2, 0, 0, 0, 2, 1, 0, 2, 0, 2, 4, 1, 1, 1, 2, 0, 1, 1,
           1, 1, 0, 2, 3, 2, 0, 2, 1, 3, 1, 3, 2, 2, 3, 2, 0, 0, 0, 1, 0, 0, 0, 1, 2, 0,
           3, 3, 0, 1, 2, 2, 2, 0, 6, 0, 0, 0, 2, 0, 1, 1, 1, 3, 3, 2, 1, 1, 0, 1, 0, 0,
           2, 0, 2, 0, 1, 0, 2, 0, 0, 2, 2, 4, 1, 2, 3, 2, 0, 0, 0, 1, 0, 0, 1, 5, 2, 1,
           3, 2, 0, 2, 1, 1, 3, 0, 5, 0, 0, 2, 4, 3, 4, 0, 0, 0, 0, 0, 0, 2, 2, 0, 0, 2,
           0, 0, 1, 1, 0, 2, 1, 3, 3, 2, 2, 0, 0, 2, 3, 2, 4, 3, 3, 4, 0, 3, 0, 1, 0, 1,
           2, 3, 4, 1, 2, 6, 2, 1, 2, 2)
sum_A <- sum(man_A)
sum_B <- sum(man_B)
n_A <- length(man_A)
n_B <- length(man_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")
```



d. The plot shows that the expectation of the difference between θ_A and θ_B decreases to 0 as a_γ, b_γ increases to infinity. This could be seen from the posterior mean of the full conditional distribution of γ, which is $\frac{n_B}{n_B\theta+b_\gamma}\frac{\sum y_{Bi}}{n_B} + \frac{b_\gamma}{n_B\theta+b_\gamma}\frac{a_\gamma}{b_\gamma}$. This converges to 1 when prior sample size $a_\gamma = b_\gamma \to \infty$. Since $\frac{\theta_B}{\theta_A} = \gamma$, $E(\theta_A|y_A, y_B) = E(\theta_B|y_A, y_B)$ and thus $E(\theta_B - \theta_A|y_A, y_B) = 0$.

Question 2:

**a) Show the exact steps involved in an algorithm for sampling from the posterior distribution for λ and γ.*8

We know from homework 3 that $p(\lambda|\gamma, x, y) = Gamma(\sum y_i + 1, n_0 + n_1\gamma + 1)$ given n_0, n_1 as number of observations when $x_i = 0$ and $x_i = 1$ respectively.

We just need to derive the full conditional of γ.

$$\begin{aligned}
p(\gamma|\lambda, x, y) &\propto p(\gamma, \lambda, x, y) \\
&\propto p(x, y|\lambda, \gamma)p(\lambda, \gamma) \\
&\propto p(x, y|\lambda, \gamma)p(\gamma) \\
&= p(y|x, \lambda, \gamma)p(\gamma) \\
&\propto \prod_{i=1}^{n} (\lambda\gamma^{x_i})^{y_i} e^{-\lambda\gamma^{x_i}} \gamma^{1-1} e^{-\gamma} \\
&= \prod_{i=1}^{n} \lambda^{y_i}\gamma^{x_i y_i} e^{-\lambda\gamma^{x_i}} e^{-\gamma} \\
&\propto \gamma^{\sum x_i y_i} e^{-\lambda(n_0+n_1\gamma)} e^{-\gamma} \\
&\propto \gamma^{\sum x_i y_i} e^{-(\lambda n_1\gamma+\gamma)} \\
&= \gamma^{\sum x_i y_i} e^{-\gamma(\lambda n_1+1)} \\
&= Gamma(\sum x_i y_i + 1, \lambda n_1 + 1)
\end{aligned}$$

Therefore, the exact steps involved in gibbs sampler are the following:

- Start with intial value λ_0 and γ_0
- For iterations t=1,…,1000

- Sample $\lambda^{(t)}$ from the conditional posterior distribution

$$p(\lambda|\gamma, x, y) = Gamma(\sum y_i + 1, n_0 + n_1\gamma^{t-1} + 1)$$

- Sample $\gamma^{(t)}$ from the conditional posterior distribution

$$p(\gamma|\lambda, x, y) = Gamma(\sum x_i y_i + 1, \lambda^{t-1} n_1 + 1)$$

- This generates a dependent sequence of parameter values.

b) Simulate data and generate samples from the posterior distribution for these data.

```
# simulate data and prior
lambda = gamma = 1
set.seed(1)
treated = rpois(n=50, lambda=1)
control = rpois(n=50, lambda=1)

sum_tr <- sum(treated)
sum_con <- sum(control)
n_tr <- length(treated)
n_con <- length(control)

S = 1000 # number of samples to draw

# starting point
lambda_0 = 1
gamma_0 = 1
gibbs_lambda = matrix(nrow = 1, ncol = S + 1)
gibbs_gamma = matrix(nrow = 1, ncol = S + 1)
gibbs_lambda[,1] = lambda_0
gibbs_gamma[,1] = gamma_0

# gibbs sampler
set.seed(1234)
for (s in 2:(S+1)){
  gibbs_lambda[,s] = rgamma(1, shape = sum_tr + sum_con + 1,
                            rate = n_con + n_tr * gibbs_gamma[,s-1] + 1)
  gibbs_gamma[,s] = rgamma(1, shape = sum_tr+1,
                            rate = n_tr * gibbs_lambda[,s] + 1)
}
```

c) Use your code to - estimate the posterior mean and a 95% credible interval for $log(\lambda)$ - estimate the predictive distribution for subjects having $x_i = 0$ and $x_i = 1$. Are these predictive distributions different?

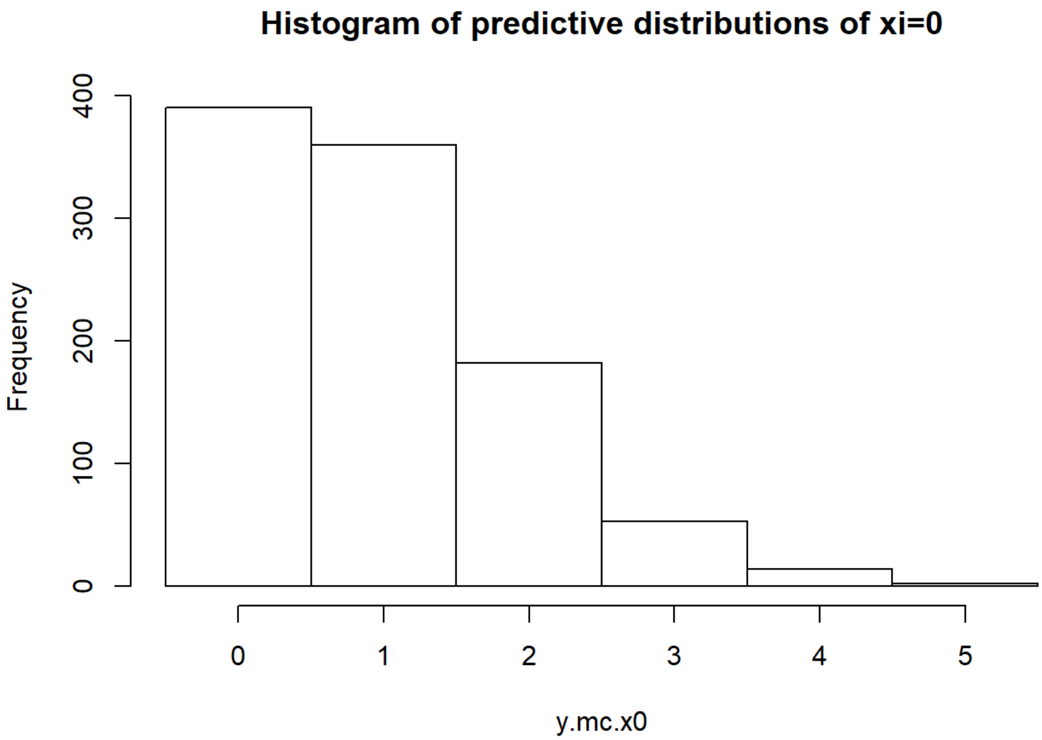
```
# posterior mean for log(lambda)
mean(log(gibbs_lambda[1,])) #-0.0721
```

```
## [1] -0.07213057
```

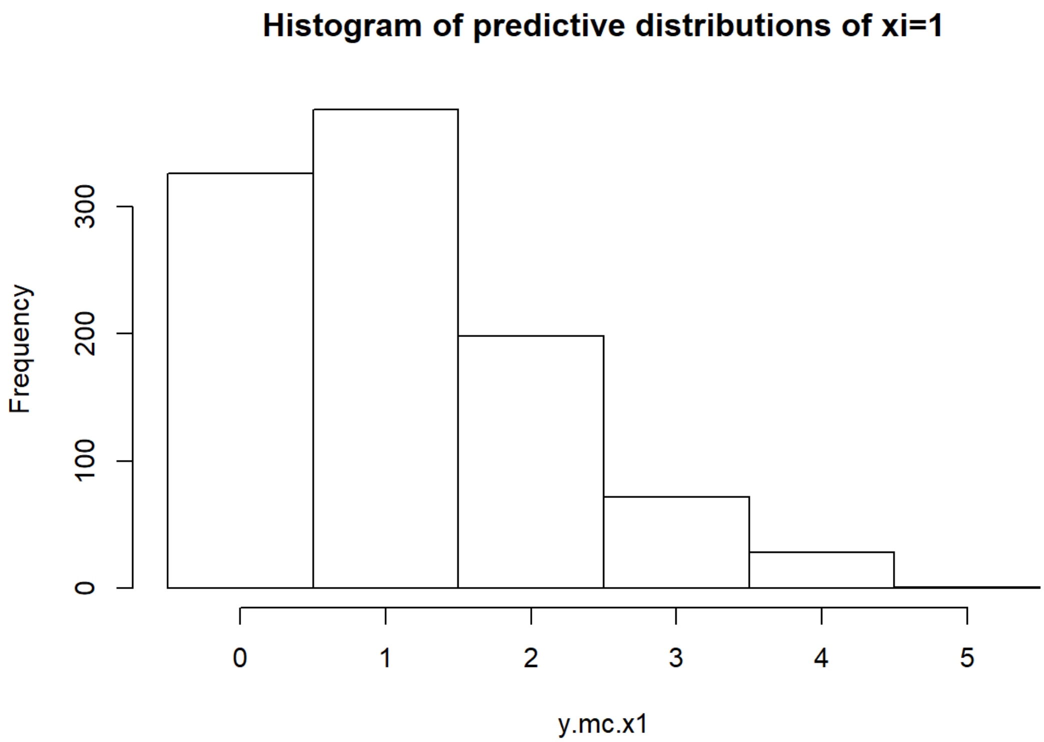
```
# 95% credible interval for log(lambda)
quantile(log(gibbs_lambda[1,]), c(0.025,0.975))
```

```
##       2.5%      97.5%
## -0.3423654  0.1977386
```

```r
# predictive distribution
#1) for x_i = 0
y.mc.x0 = rpois(1001, gibbs_lambda[1,])
hist(y.mc.x0, breaks = seq(-0.5, max(y.mc.x0)+0.5,1), main="Histogram of predictive distributions of xi=0")
```




```r
#2) for x_i = 1
y.mc.x1 = rpois(1001, gibbs_lambda[1,]*gibbs_gamma[1,])
hist(y.mc.x1, breaks = seq(-0.5, max(y.mc.x1)+0.5,1), main="Histogram of predictive distributions of xi=1")
```



Yes, these predictive distributions are slightly different in terms of the frequency of 0 and 1. When $x_i = 0$, the frequency of 0 is the highest, while $x_i = 1$, the frequency of 1 is the highest.

*8d) Run convergence diagnostics - is your chain mixing well? What is the effective sample size? Does the mixing differ for λ and γ?**

```
library(truncnorm)
library(coda)

# summary statistics
summary(mcmc(gibbs_lambda[1,], start = 1))
```

```
##
## Iterations = 1:1001
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 1001
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##           Mean             SD       Naive SE Time-series SE
##       0.939502       0.131277       0.004149       0.007157
##
## 2. Quantiles for each variable:
##
##   2.5%    25%    50%    75%  97.5%
## 0.7101 0.8469 0.9324 1.0213 1.2186
```

```
summary(mcmc(gibbs_gamma[1,], start = 1))
```

```
##
## Iterations = 1:1001
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 1001
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##           Mean             SD       Naive SE Time-series SE
##       1.177383       0.230249       0.007277       0.013213
##
## 2. Quantiles for each variable:
##
##   2.5%    25%    50%    75%  97.5%
## 0.7888 1.0130 1.1591 1.3158 1.6862
```

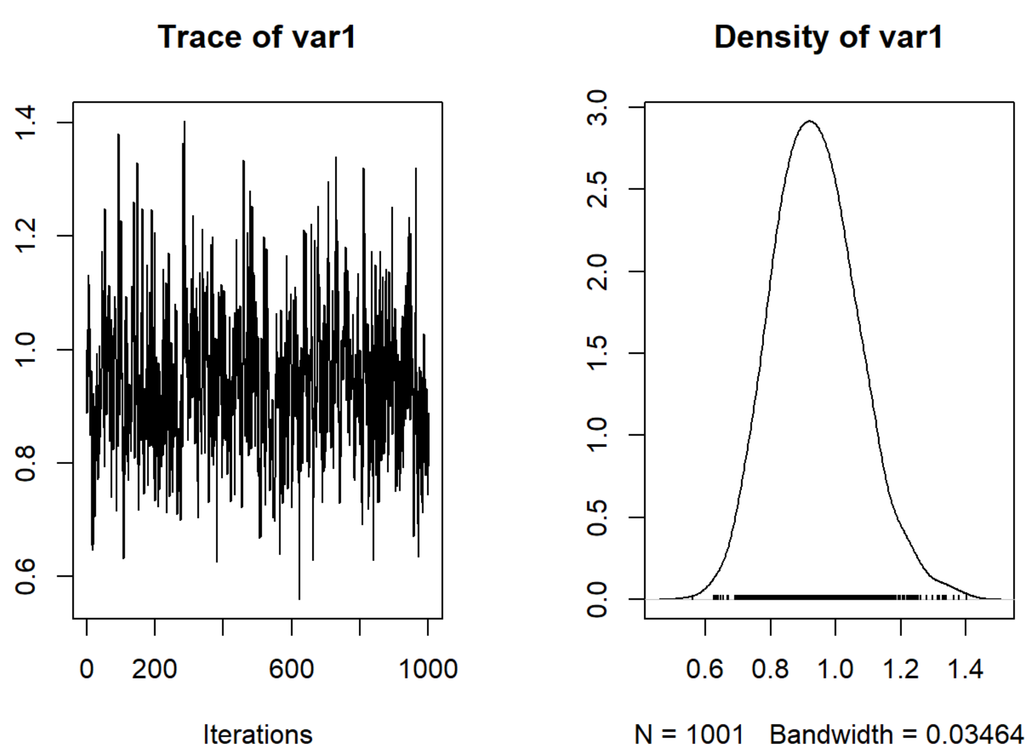
```
# effective sample size
effectiveSize(mcmc(gibbs_lambda[1,], start = 1))
```

```
##     var1
## 336.4329
```

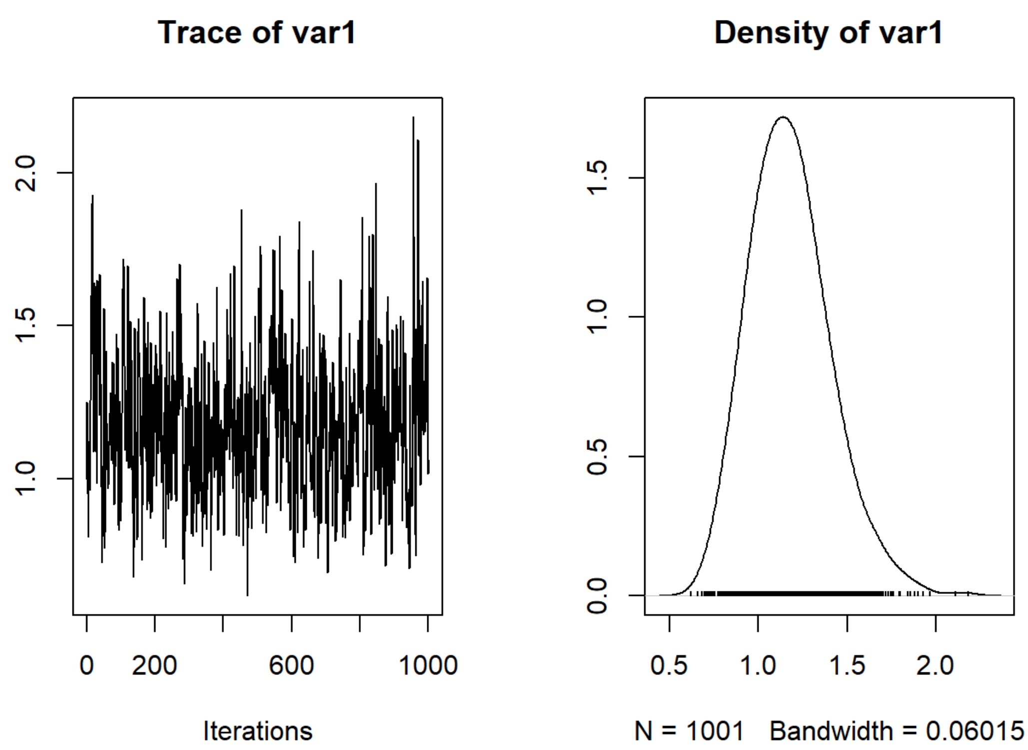
```
effectiveSize(mcmc(gibbs_gamma[1,], start = 1))
```

```
##     var1
## 303.6749
```

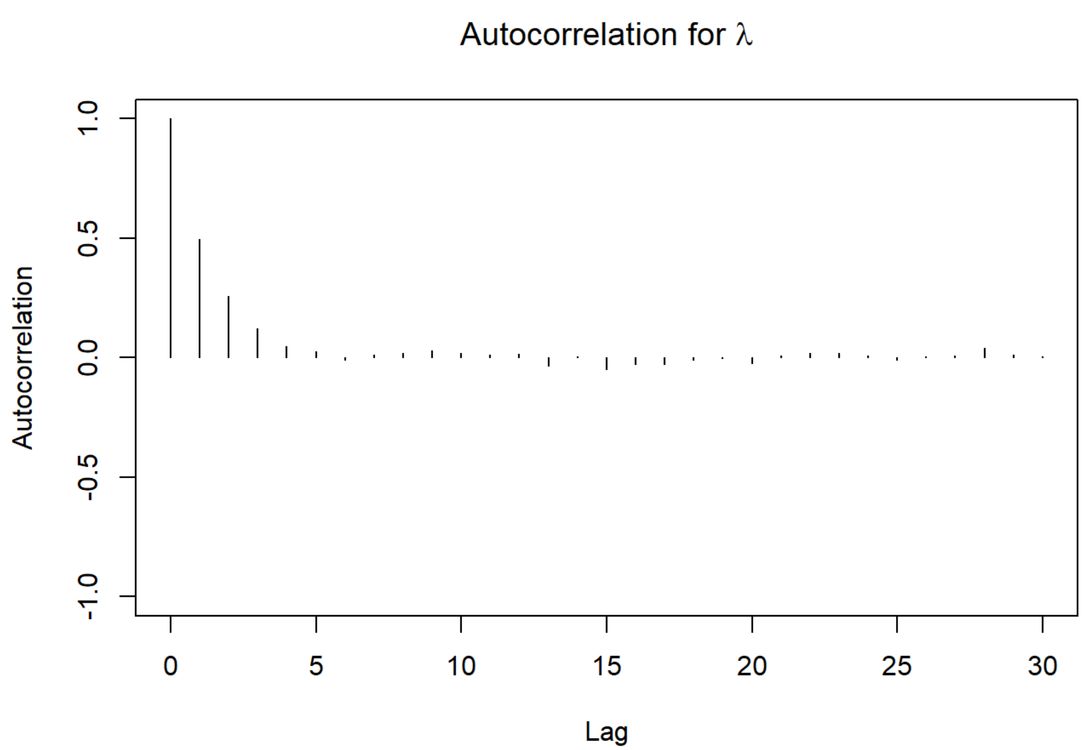
```
# trace plots
plot(mcmc(gibbs_lambda[1,], start = 1))
```



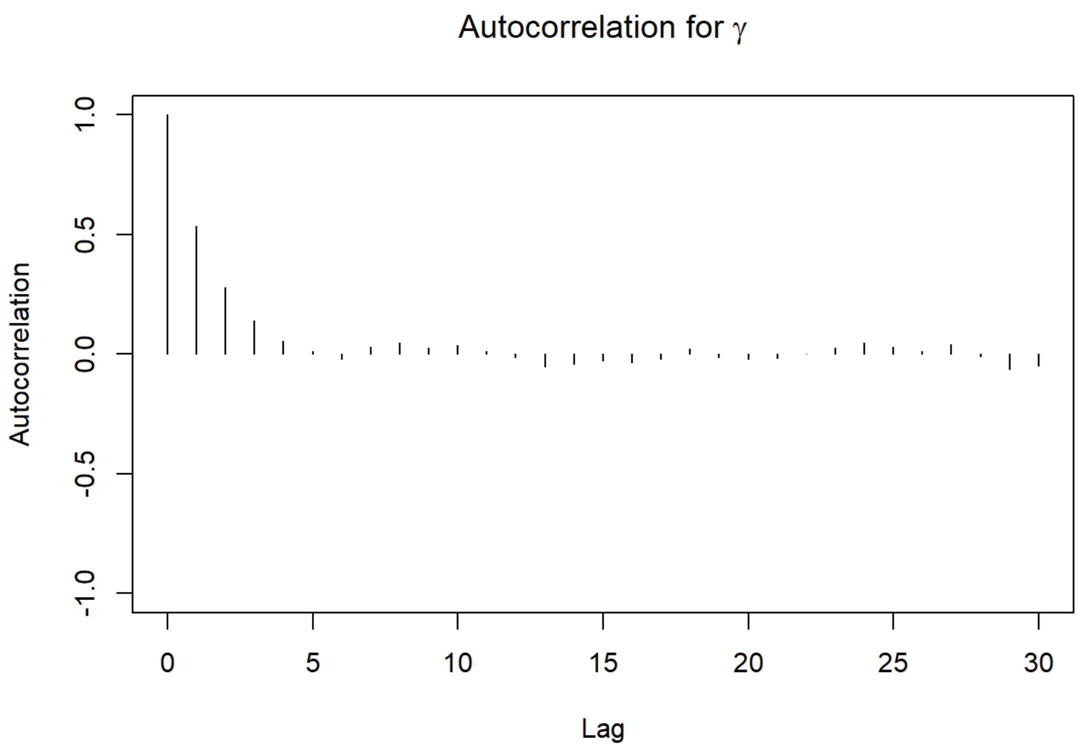
```
plot(mcmc(gibbs_gamma[1,], start = 1))
```




```
# autocorrelation plot
autocorr.plot(mcmc(gibbs_lambda[1,]), main=expression(paste("Autocorrelation for ",lambda)))
```




```
autocorr.plot(mcmc(gibbs_gamma[1,]), main=expression(paste("Autocorrelation for ",gamma)))
```



From the trace plots of both λ and γ, we can see that there isn't cyclic local trends in the mean. Also from the autocorrelation plots, we can see that after 6-7 lags, the autocorrelation dies out,which means a pretty good mixing from observation.