Winter 18 – AMS206B Homework 7 Solution

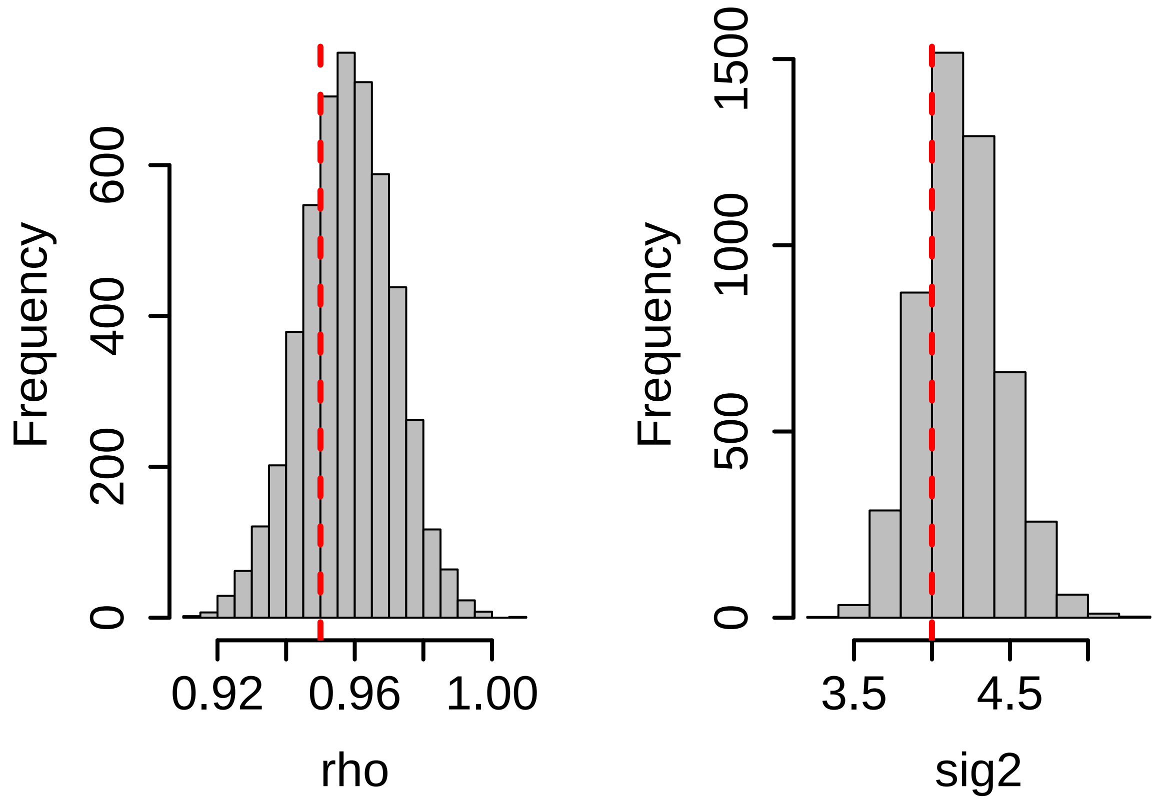
1. Let $y_t = \rho y_{t-1} + \epsilon_t$, $\epsilon_t \overset{i.i.d}{\sim} N(0, \sigma^2)$. This is a popular model in time series analysis known as the autoregressive model of order one or AR(1).

   In hw5, we assumed a prior of the form $\pi(\rho, \sigma^2) \propto 1/\sigma^2$, and found $p(\rho, \sigma^2|y_1, \ldots, y_n)$, $p(\rho|\sigma^2, y_1, \ldots, y_n)$ and $p(\sigma^2|y_1, \ldots, y_n)$ based on the conditional likelihood.

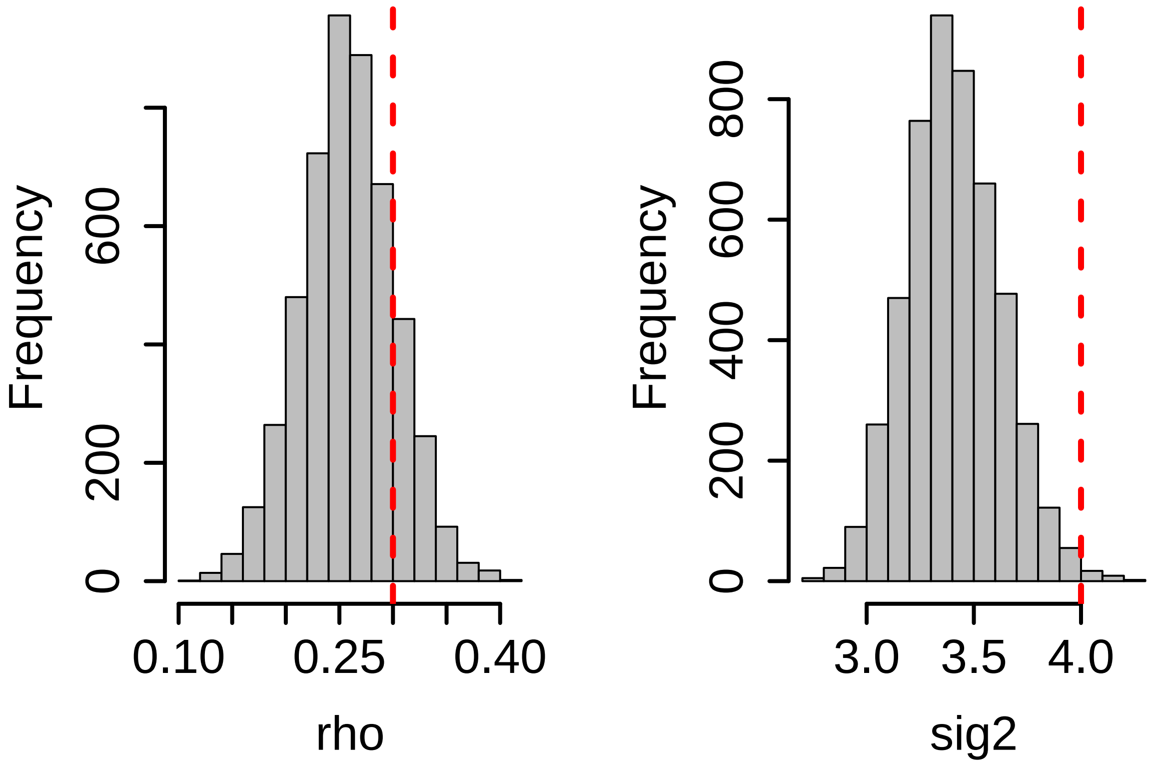
   Simulate two data sets with $n = 500$ observations each. One with $\rho = 0.95, \sigma^2 = 4$ and another one with $\rho = 0.3, \sigma^2 = 4$. Fit the model above to the two data sets. Summarize your posterior results in both cases.

Solution:

```r
# set the number of observations and the true value of the parameters
n <- 500
tr.rho <- 0.95
tr.sig2 <- 4

# generate data
y <- rep(NA, n)
y[1] <- rnorm(1, 0, sqrt(tr.sig2))
for(i in 2:n) y[i] <- rnorm(1, tr.rho*y[i-1], sqrt(tr.sig2))

# compute statistics to be used in the parameters of the posterior
sum.y2 <- sum(y[-1]^2)
sum.yy <- sum(y[-1]*y[-n])

# set the number of Monte Carlo samples
N.sam <- 5000

# use direct sampling to obtain samples from the joint posterior by
# sampling from the marginal posterior for sig2 and, then, use those
# samples to generate samples from the conditional posterior for rho
sig2 <- 1/rgamma(N.sam,(n/2 - 1), (sum.y2 - (sum.yy)^2/sum.y2)/2)
rho <- rnorm(N.sam , sum.yy/sum.y2, sqrt(sig2/sum.y2))

# adjust graphical parameters
par(mar=c(4.5, 4.5, 2.1, 2.1), mfrow=c(1,2))

# plot histogram of the rho samples
hist(rho, col=8, lwd=2, , main="", cex.axis=1.5, cex.lab=1.5)

# add vertical line at the true value of rho.
abline(v=tr.rho, lty=2, lwd=3, col=2)

# repeat ploting for sig2
hist(sig2, col=8, lwd=2, , main="", cex.axis=1.5, cex.lab=1.5)
abline(v=tr.sig2, lty=2, lwd=3, col=2)
```




```r
# repeat for rho=.3
tr.rho <- 0.3

y <- rep(NA, n)
y[1] <- rnorm(1, 0, sqrt(tr.sig2))
for(i in 2:n) y[i] <- rnorm(1, tr.rho*y[i-1], sqrt(tr.sig2))

sum.y2 <- sum(y[-1]^2)
sum.yy <- sum(y[-1]*y[-n])

sig2 <- 1/rgamma(N.sam,(n/2 - 1), (sum.y2 - (sum.yy)^2/sum.y2)/2)
rho <- rnorm(N.sam , sum.yy/sum.y2, sqrt(sig2/sum.y2))

par(mar=c(4.5, 4.5, 2.1, 2.1), mfrow=c(1,2))

hist(rho, col=8, lwd=2, , main="", cex.axis=1.5, cex.lab=1.5)
abline(v=tr.rho, lty=2, lwd=3, col=2)

hist(sig2, col=8, lwd=2, , main="", cex.axis=1.5, cex.lab=1.5)
abline(v=tr.sig2, lty=2, lwd=3, col=2)
```



From this figures it is clear that the model reasonably recovers the true values of ρ and σ^2 and that the estimates of ρ are closer to their true values in both scenarios.

2. Consider a model of the form $x \mid \theta \sim \mathcal{B}in(n, \theta)$ and $\theta \sim \mathcal{B}eta(1/2, 1/2)$. Assume that you observe $n = 10$ and $x = 1$.

   (a) Report an exact 95% (symmetric) posterior credible interval for þ (for example, you can use the `qbeta` function in `R`).
   (b) Report an approximate credible interval for θ using the Laplace approximation.
   (c) Report an approximate credible interval for θ using Monte Carlo simulation.
   (d) Repeat the previous calculations with $n = 100, x = 10$ and with $n = 1000, x = 100$. Comment on the difference between all 9 situations.

Solution:

(a) In this case the posterior is given by $\theta \mid x \sim \mathcal{B}eta(a = 3/2, b = 19/2)$ and the exact posterior credible interval can be calculated as

```
a<-3/2
b<-19/2
c(qbeta(.025,a,b),qbeta(.975,a,b))
```

```
## [1] 0.01101167 0.38131477
```

(b) Using the Laplace approximation

$$
\begin{aligned}
P(\ell_1 < \theta < \ell_2 \mid x) &= \int_{\ell_1}^{\ell_2} \frac{1}{B(a,b)} \theta^{a-1}(1-\theta)^{b-1} d\theta \\
&= \frac{1}{B(a,b)} \int_{\ell_1}^{\ell_2} \exp\{(a-1)\log(\theta) + (b-1)\log(1-\theta)\} d\theta \\
&\approx \frac{1}{B(a,b)} \exp\{h(\hat\theta)\} \sqrt{\frac{2\pi}{-h''(\hat\theta)}} \int_{\ell_1}^{\ell_2} \sqrt{\frac{-h''(\hat\theta)}{2\pi}} \exp\left\{-\frac{|h''(\hat\theta)|(\theta-\hat\theta)^2}{2}\right\} d\theta \\
&\approx \frac{1}{B(a,b)} \exp\{h(\hat\theta)\} \sqrt{\frac{2\pi}{-h''(\hat\theta)}} \left\{\Phi\left(\sqrt{-h''(\hat\theta)}(\ell_2-\hat\theta)\right) - \Phi\left(\sqrt{-h''(\hat\theta)}(\ell_1-\hat\theta)\right)\right\},
\end{aligned}
$$

where $h(\theta) = (a-1)\log(\theta) + (b-1)\log(1-\theta)$ and $\hat\theta$ is such that $h'(\hat\theta) = 0$.

Taking first and second derivatives, $h(\hat\theta) = (a-1)\log(\hat\theta) + (b-1)\log(1-\hat\theta)$ and $h''(\hat\theta) = -\frac{a-1}{\hat\theta^2} - \frac{b-1}{(1-\hat\theta)^2}$, yielding $\hat\theta = (a-1)/(a+b-2)$. Hence the interval can be computed as follows

```
# calculate the MAP
th.hat <- (a-1)/(a+b-2)

# evaluate the first and second derivaties at the MAP
h <- (a-1)*log(th.hat) + (b-1)*log(1-th.hat)
h.2 <- -(a-1)/th.hat^2 - (b-1)/(1-th.hat)^2

#calculate the constant
Const <- exp(h)*sqrt(2*pi/(-h.2))/beta(a, b)

# compute the interval
lower<-th.hat + qnorm((1 - 0.95/Const)/2, 0, 1)/sqrt(-h.2)
upper<-th.hat + qnorm((1 - 0.95/Const)/2, 0, 1, lower.tail=FALSE)/sqrt(-h.2)

c(lower,upper)
```

```
## [1] -0.1736783  0.2847894
```

(c) Using Monte Carlo simulation, the interval is found by obtainning a sample (of size 2000) from the posterior and looking at the empirical quantiles.

```
th <- rbeta(2000, a, b)
quantile(th, probs=c(0.025, 0.975))
```

```
##       2.5%      97.5%
## 0.01239312 0.38493013
```

(d) Table 1 shows the 95% credible intervals for the three cases; (i) $n = 10$ and $x = 1$, (ii) $n = 100$ and $x = 10$ and (iii) $n = 1000$ and $x = 100$. Note that all the three cases have $x/n = 0.1$ and the prior is vague (in fact, it is the Jeffereys prior for the binomial distribution). As n increase, the posterior distribution becomes centered at 0.1 more sharply. Thus, the approximate intervals, especially the interval by the Laplace approximation, are improved significantly. Since we have a fairly large MC sample size, the approximate intervals by MC simulation are accurate.

3. Let $x_1, \dots, x_n$ be an i.i.d. sample such that $x_i|\theta, \sigma^2 \sim \mathcal{N}(\theta, \sigma^2)$ with θ and σ^2 unknown. Assume a conjugate normal-inverse-gamma prior on (θ, σ^2) such that $\theta|\sigma^2 \sim \mathcal{N}(\theta_0, \kappa_0\sigma^2)$ and $\sigma^2 \sim IG(a, b)$ with θ_0, κ_0, a and b known.

In hw6 we found that $p(\theta, \sigma^2|x_1, \dots, x_n)$, $p(\theta|\sigma^2, x_1, \dots, x_n)$, $p(\sigma^2|x_1, \dots, x_n)$ and $p(\theta|x_1, \dots, x_n)$.

n	x	Exact	Laplace Approx.	Monte Carlo
10	1	(0.011, 0.381)	(-0.174, 0.285)	(0.012, 0.393)
100	10	(0.053, 0.170)	(0.038, 0.154)	(0.052, 0.172)
1000	100	(0.083, 0.120)	(0.081, 0.118)	(0.082, 0.120)

Table 1: [Q2] 95% credible intervals using three methods.

(a) Simulate $n = 1000$ i.i.d. observations from a $N(5, 1)$. Fit the above model to these data assuming the following prior scenarios:

i. fairly informative priors around the true values of both parameters
ii. informative prior on θ and vague on σ^2
iii. informative prior on σ^2 and vague on θ
iv. vague on both parameters

Specify the form of your posteriors in each case.

(b) Assume that you are interested in estimating $\eta = \theta/\sigma$. Develop a Monte Carlo algorithm for computing the posterior mean and a 95% credibility interval for η. Use the algorithm to compute such quantities under all the prior scenarios described above.

Solution:

(a) As found in hw6, the posterior in this case satisfies $\theta \mid \sigma^2, \mathbf{x} \sim \mathcal{N}\left(m, v^2\right)$ and $\sigma^2 \mid \mathbf{x} \sim \mathcal{IG}(a', b')$

where $m = \frac{\kappa_0 \sigma^2}{1+n\kappa_0}\left(\frac{\theta_0}{\kappa_0\sigma^2} + \frac{\sum x_i}{\sigma^2}\right)$, $v^2 = \frac{\kappa_0\sigma^2}{1+n\kappa_0}$, $a' = a + n/2$ and $b' = \left[b^{-1} + \frac{1}{2}\sum_{i=1}^n (x_i - \bar{x})^2 + \frac{n(\bar{x}-\theta_0)^2}{2(1+n\kappa_0)}\right]^{-1}$

Then, to simulate θ and σ^2 from the joint posterior distribution we use the following algorithm:

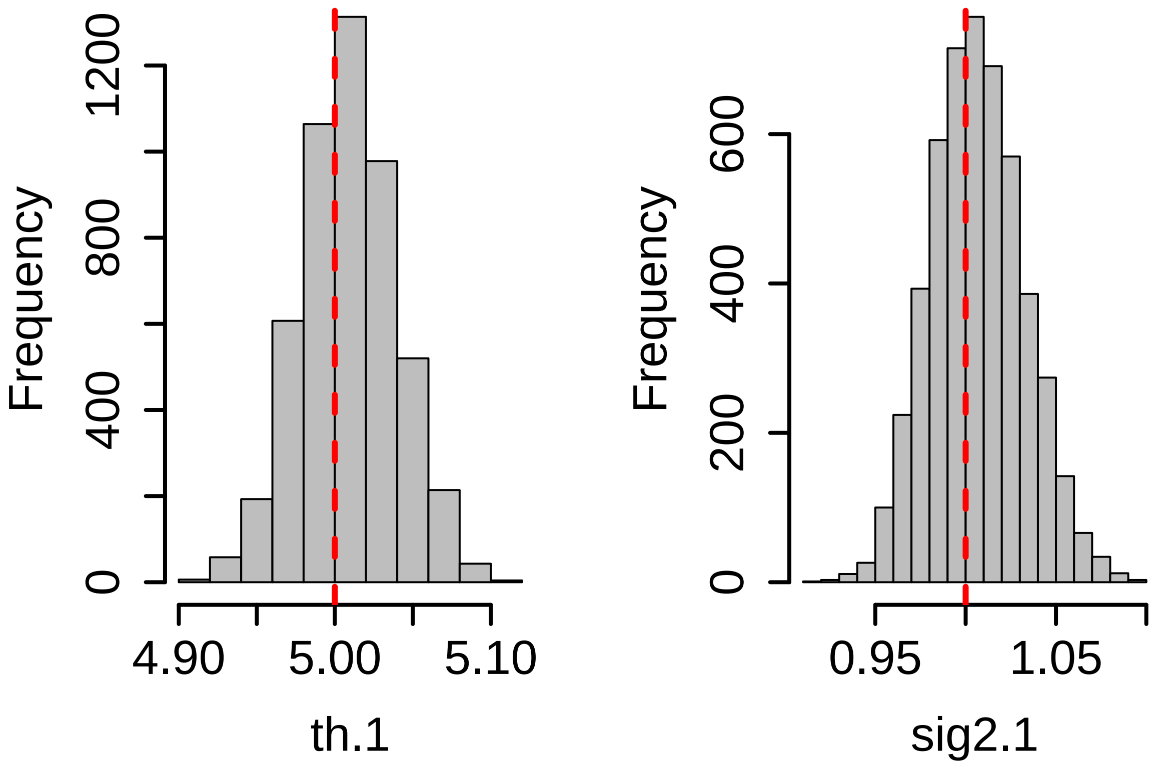
step 1: Generate σ^2 from $f(\sigma^2 \mid \mathbf{x})$

step 2: Generate θ from $f(\theta \mid \sigma^2, \mathbf{x})$.

which is implemented as

```
# set the number of observations and true value of parameters
n <- 1000
tr.th <- 5
tr.sig2 <- 1

# generate dataset
x <- rnorm(n, tr.th, sqrt(tr.sig2))

#set the number of MC samples
N.sam <- 5000

#i. set hyperparameters for fairly informative priors
th0 <- tr.th
k0 <- 0.01
a <- 1001
b <- tr.sig2/(a-1)

#calculate posterior parameters
m <- (th0 + n*k0*mean(x))/(1+n*k0)
alpha <- a + n/2
beta <- 1/b + sum(x^2)/2 + th0^2/(2*k0) - (th0 + n*k0*mean(x))^2/(2*k0*(1+n*k0))

#MC simulation
```

```r
sig2.1 <- 1/rgamma(N.sam, alpha, beta)
th.1 <- rnorm(N.sam, m, sqrt(sig2.1/(1/k0 + n)))

par(mar=c(4.5, 4.5, 2.1, 2.1), mfrow=c(1,2))

hist(th.1, col=8, lwd=2, , main="", cex.axis=1.5, cex.lab=1.5)
abline(v=tr.th, lty=2, lwd=3, col=2)

hist(sig2.1, col=8, lwd=2, , main="", cex.axis=1.5, cex.lab=1.5)
abline(v=tr.sig2, lty=2, lwd=3, col=2)
```




```r
# for ii.-iv. repeat the above code changing the value of the hyperparameters accordingly.
```

Table 2 shows different combinations of the hyperparameters yielding the desired form of the priors.

(b) An advantage of MC methods is that computing the distribution (or a summary) of a transformed parameter is straightforward. In this case:

```r
# posterior mean of eta
mean(th.1/sqrt(sig2.1))
```

```
## [1] 4.993593
```

```r
# 95% credible interval of eta
quantile(th.1/sqrt(sig2.1), probs=c(0.025, 0.975))
```

```
##     2.5%    97.5%
## 4.853991 5.130723
```

Table 2 shows the posterior means and 95% credible intervals for η under the distinct choices of hyperparameters. As can be seen there, because the sample size ($n = 1000$) is fairly large, the the prior distribution does not have a huge impact on the posterior. Having said that, the posterior mean of η is close to its true value, 5, under the four different prior calibration. Interestingly, when the prior distribution on σ^2 is informative, the credible interval is shorter. This is due to the fact that σ^2 also controls the dispersion of the prior of θ.

	θ_0	κ_0	a	b	post. mean	95% CI
i	5	0.01	1001	0.001	5.04	(4.903, 5.185)
ii	5	0.01	3	0.5	5.066	(4.837, 5.296)
iii	5	100	1001	0.001	5.04	(4.901, 5.185)
v	5	100	3	0.5	5.07	(4.846, 5.293)

Table 2: [Q3] Hyperparameter values and posterior inference.

4. (Wasserman, 2003) A random variable Z has an *inverse Gaussian distribution* if it has density

$$f(z \mid \theta_1, \theta_2) \propto z^{-3/2} \exp\left\{-\theta_1 z - \frac{\theta_2}{z} + 2\sqrt{\theta_1\theta_2} + \log(\sqrt{2\theta_2})\right\}, \quad z > 0,$$

where $\theta_1 > 0$ and $\theta_2 > 0$ are parameters. It can be shown that $E(Z) = \sqrt{\theta_2/\theta_1}$ and also that $E(1/Z) = \sqrt{\theta_1/\theta_2} + 1/(2\theta_2)$.

(a) Let $\theta_1 = 1.5$ and $\theta_2 = 2$. Draw a sample of size 1,000 using the independence-Metropolis-Hastings method with a Gamma distribution as the proposal density (note that in an independence-Metropolis-Hastings $q(\theta^*|\theta) = q(\theta^*)$). To assess the accuracy of the method, compare the mean of Z and $1/Z$ from the sample to the theoretical means. Try different Gamma distributions to see if you can get an accurate sample.

(b) Draw a sample of size 1,000 using the random-walk Metropolis method. Since $z > 0$ we cannot just use a Normal density. Let $W = \log(Z)$. Find the density of W. Use the random-walk Metropolis method to get a sample $W_1, \ldots, W_M$ and let $Z_i = e^{W_i}$. Assess the accuracy of the simulation as in the previous part.

Solution:

(a) Using the values of θ_1 and θ_2, the theoretical means of Z and $1/Z$ are calculated to be 1.154701 and 1.404701 respectively.

As a first step is possible to use a $\mathcal{G}a(115.4701, 100)$, so that the mean of the proposal distribution matches the mean of the theoretical distribution with relatively small variance.

```
# set number of M-H samples and true value of the parameters
N.sam <- 1000
th1 <- 1.5
th2 <- 2

# calculate teorethical means of Z and 1/Z
sqrt(th2/th1)
```

```
## [1] 1.154701
```

```
sqrt(th2/th1) + 1/(2*th2)
```

```
## [1] 1.404701
```

```
# set hyperparameters
b <- 100
a <- sqrt(th2/th1)*b

# initialize chain
MH.indep <- rep(NA, N.sam)
```

```r
accpt.cnt <- 0
z.cur <- 1.0

# do M-H
for(i.sam in 1:N.sam)
{
    # evaluate the pdf at the current value of z
    p.cur <- -(3/2)*log(z.cur) - th1*z.cur - th2/z.cur

    # generate a proposed value
    z.pro <- rgamma(1, a, b)

    # evaluate the pdf at the proposed value of z
    p.pro <- -(3/2)*log(z.pro) - th1*z.pro - th2/z.pro

    # calculate acceptance probability
    accpt.prob <- exp(p.pro + dgamma(z.cur,a,b,log=T) - p.cur - dgamma(z.pro,a,b,log=T))

    # accept or reject acordingly
    if(runif(1) < accpt.prob)
    {
        z.cur <- z.pro
        accpt.cnt <- accpt.cnt + 1
    }
    MH.indep[i.sam] <- z.cur

}

# calculare MC estimates of the mean of Z and 1/Z
mean(MH.indep)
```

```
## [1] 1.201901
```

```r
mean(1/MH.indep)
```

```
## [1] 0.8542182
```

```r
# acceptance rate
accpt.cnt/N.sam
```

```
## [1] 0.545
```

Keeping the mean constant but increasing the disperssion of the proposal with a $\mathcal{G}a(11.54701, 10)$, the MC estimates of $\mathbb{E}[Z]$ and $\mathbb{E}[1/Z]$ are 1.056 and 1.022 respectively; while the acceptance rate changes to 80.5%. Note that this proposal distribution proposes more values away from the mean, which explains the decrease in the acceptance rate. However, the tails of the inverse Gaussian distribution are better represented in this MC sample, resulting in a better estimate of $\mathbb{E}[1/Z]$.

(b) Using that the density of W is given by

$$f_W(w) = f_Z(z(w)) \left| \frac{dz}{dw} \right| \propto \exp\left\{ -\frac{3}{2}w - \theta_1 \exp\{w\} - \frac{\theta_2}{\exp\{w\}} \right\} \exp\{w\}$$

a Random walk M-H on the logarithm scale can be implemented as follws:

```r
v <- 0.04
MH.RW <- rep(NA, N.sam)
```

```
accpt.cnt <- 0
z.cur <- 1.0

for(i.sam in 1:N.sam)
{
    p.cur <- -(1/2)*log(z.cur) - th1*z.cur - th2/z.cur
    z.pro <- exp(log(z.cur) + rnorm(1, 0, sqrt(v)))
    p.pro <- -(1/2)*log(z.pro) - th1*z.pro - th2/z.pro

    accpt.prob <- exp(p.pro - p.cur)
    if(runif(1) < accpt.prob)
    {
        z.cur <- z.pro
        accpt.cnt <- accpt.cnt + 1
    }

    MH.RW[i.sam] <- z.cur
}

mean(MH.RW)
```

```
## [1] 1.161991
```

```
mean(1/MH.RW)
```

```
## [1] 1.107076
```

```
accpt.cnt/N.sam
```

```
## [1] 0.889
```

5. Consider i.i.d. data $x_1, \ldots, x_n$ such that $x_i|\nu, \theta \sim \mathcal{G}a(\nu, \theta)$ where $E(x_i) = \nu/\theta$, and assign priors $\nu \sim \mathcal{G}a(3, 1)$ and $\theta \sim \mathcal{G}a(2, 2)$.

(a) Develop a Metropolis-within-Gibbs algorithm to sample from $p(\nu, \theta|x_1, \ldots, x_n)$ using the full conditional distributions $p(\theta|\nu, x_1, \ldots, x_n)$ and $p(\nu|\theta, x_1, \ldots, x_n)$. For the second full conditional, use a random walk proposal on $\log(\nu)$.

(b) Develop a Metropolis-Hastings algorithm that jointly proposes $\log(\nu)$ and $\log(\theta)$ using a Gaussian random walk centered on the current value of the parameters. Tune the variance-covariance matrix of the proposal using a test run that proposes the parameters independently (but evaluates acceptance jointly).

(c) Develop a Metropolis algorithm that jointly proposes $\log(\nu)$ and $\log(\theta)$ using independent proposals based on the Laplace approximation of the posterior distribution of $\log(\nu)$ and $\log(\theta)$.

(d) Develop a Metropolis algorithm that jointly proposes $\log(\nu)$ and $\log(\theta)$ using independent proposals based on a modified version of the Laplace approximation of the posterior distribution of $\log(\nu)$ and $\log(\theta)$ in which the normal distribution is replaced by a heavy tailed distribution (such as a multivariate Cauchy).

(e) Run each of the algorithms for the dataset in *hw6-5.dat* and compute the effective sample sizes associated with each parameter under each of the samplers. Also, construct trace and autocorrelation plots. Report posterior means for each of the parameters of interest, along with 95% symmetric credible intervals. Discuss.

Solution:

To be generic, let $\nu \sim \mathcal{G}a(a, b)$ and $\theta \sim \mathcal{G}a(\alpha, \beta)$. The joint posterior distribution of θ and ν satisfies

$$p(\theta, \nu \mid \mathbf{x}) \quad \propto \quad \prod_{i=1}^{n} \frac{\theta^{\nu}}{\Gamma(\nu)} x_i^{\nu-1} e^{-\theta x_i} \nu^{a-1} e^{-b\nu} \theta^{\alpha-1} e^{-\beta\theta} = \frac{(\prod_{i=1}^{n} x_i)^{\nu-1} \nu^{a-1} e^{-b\nu}}{(\Gamma(\nu))^n} \theta^{\alpha+n\nu-1} e^{-\theta(\beta+\sum_{i=1}^{n} x_i)}.$$

Now, the full conditional for θ

$$p(\theta \mid \nu, \mathbf{x}) \propto \theta^{\alpha+n\nu-1} e^{-\theta(\beta+\sum_{i=1}^{n} x_i)},$$

which is a $\mathcal{G}a(n\nu+\alpha, \beta+\sum_{i=1}^{n} x_i)$; while the full conditional for ν satisfies

$$p(\nu \mid \theta, \mathbf{x}) \propto \theta^{n\nu} \frac{(\prod_{i=1}^{n} x_i)^{\nu-1} \nu^{a-1} e^{-b\nu}}{(\Gamma(\nu))^n}$$

which is not a member of a known family of distributions.

```
# read data and calculate statistics
x <- read.table("hw6-5.txt", header=FALSE)[,1]
n <- length(x)
sum.x <- sum(x)
sum.log.x <-sum(log(x))

# call the library coda for autocorrelation plots
library(coda)

# call the library for the multivariate normal
library(mvtnorm)

# set up
sam <- NULL
sam$th <-rep(NA, N.sam)
sam$nu <-rep(NA, N.sam)
N.sam <- 5000
N.test<-500
```

(a)

```
# set variance of the proposal
v <- 0.04

# initial values
th.cur <- 2
nu.cur <- 3

for(i.sam in 1:N.sam)
{
    # update theta using Gibbs step
    th.cur <- rgamma(1, n*nu.cur + alpha, beta + sum.x)

    # get proposed value for nu using a Random Walk proposal on log scale
    nu.pro <- exp(log(nu.cur) + rnorm(1, 0, sqrt(v)))

    # evaluate target pdf at current value
    p.nu.cur <- nu.cur*(sum.log.x+n*log(th.cur)-1) - n*lgamma(nu.cur) + 3*log(nu.cur)

    # evaluate target pdf at proposed value
    p.nu.pro <- nu.pro*(sum.log.x+n*log(th.cur)-1) - n*lgamma(nu.pro) + 3*log(nu.pro)
```

```r
    # calculate acceptance probability and accept/reject accordingly
    accpt.prob <- exp(p.nu.pro - p.nu.cur)
    if(runif(1) < accpt.prob)
    {
        nu.cur <- nu.pro
    }

    # save current draw
    sam$th[i.sam] <- th.cur
    sam$nu[i.sam] <- nu.cur
}

# find effective sample size
effectiveSize(sam$th)
```

```
##     var1
## 4132.713
```

```r
effectiveSize(sam$nu)
```

```
##     var1
## 1064.151
```

```r
# find posterior means
mean(sam$nu)
```

```
## [1] 2.485038
```

```r
mean(sam$th)
```

```
## [1] 0.994784
```

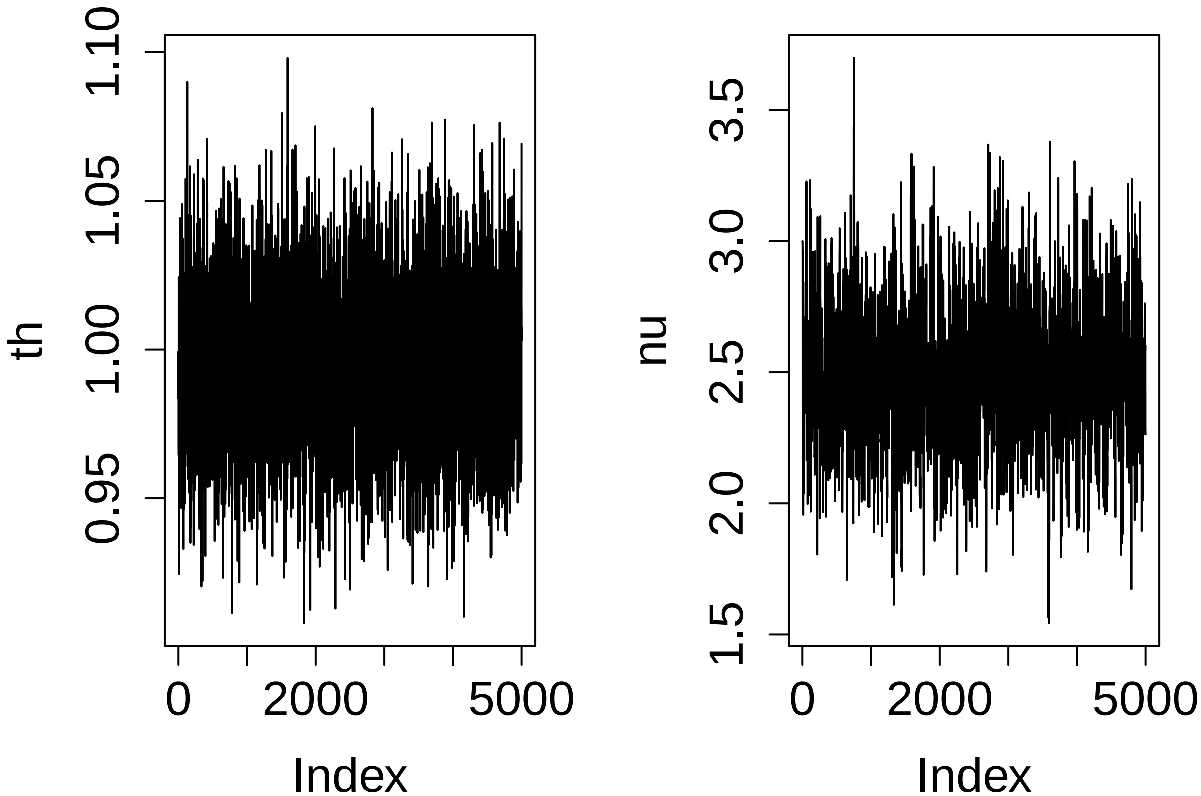
```r
# find the 95% credible intervals
quantile(sam$nu, probs=c(0.025, 0.975))
```

```
##     2.5%    97.5%
## 1.960581 3.033201
```

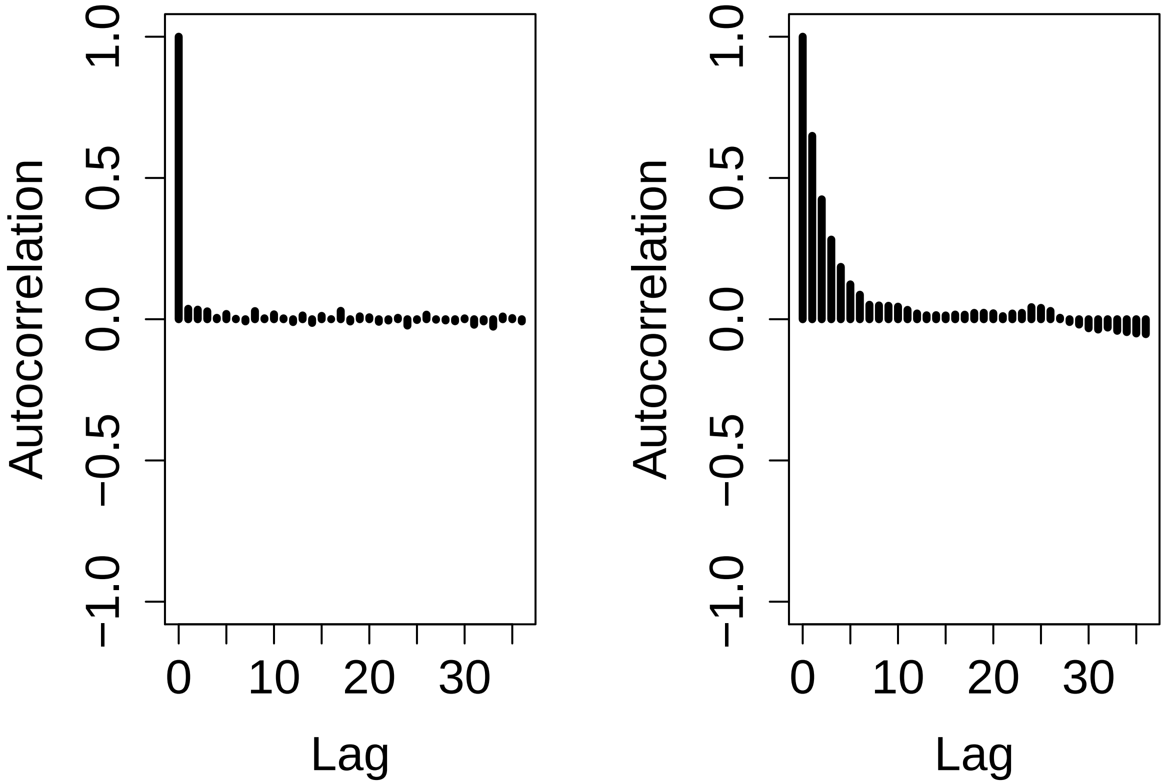
```r
quantile(sam$th, probs=c(0.025, 0.975))
```

```
##      2.5%     97.5%
## 0.9454839 1.0465900
```

```r
# traceplots
par(mfrow=c(1,2), mar=c(4.5, 4.5, 2.1, 2.1))
plot(sam$th, col=1, type="l", main="", cex.axis=1.5, cex.lab=1.5, ylab="th")
plot(sam$nu, col=1, type="l", main="", cex.axis=1.5, cex.lab=1.5, ylab="nu")
```




```r
# autocorrelation plots
par(mfrow=c(1,2), mar=c(4.5, 4.5, 2.1, 2.1))
autocorr.plot(sam$th, col=1, lwd=4, cex.axis=1.5, cex.lab=1.5, auto.layout = FALSE, main="")
autocorr.plot(sam$nu, col=1, lwd=4, cex.axis=1.5, cex.lab=1.5, auto.layout = FALSE, main="")
```



(b)

```
# set variance of the proposal for test run
V <- 0.04 * diag(2)

# initial values
th.cur <- 2
nu.cur <- 3

for (i.sam in 1:N.test) {
    # propose (nu, theta) jointly from a Random Walk
    nu.pro <- exp(log(nu.cur) + rnorm(1, 0, sqrt(V[1, 1])))
    th.pro <- exp(log(th.cur) + rnorm(1, 0, sqrt(V[2, 2])))

    # evaluate the joint target density (on the log scale) at the current value
    p.cur <- (n * nu.cur + 2) * log(th.cur) - n * lgamma(nu.cur) + nu.cur * (sum.log.x -
        1) + 3 * log(nu.cur) - th.cur * (2 + sum.x)

    # evaluate the joint target density (on the log scale) at the proposed value
    p.pro <- (n * nu.pro + 2) * log(th.pro) - n * lgamma(nu.pro) + nu.pro * (sum.log.x -
        1) + 3 * log(nu.pro) - th.pro * (2 + sum.x)

    # calculate acceptance probability and accept/reject accordingly
    accpt.prob <- exp(p.pro - p.cur)
    if (runif(1) < accpt.prob) {
        nu.cur <- nu.pro
        th.cur <- th.pro
    }
```

```r
    # save current draw
    sam$th[i.sam] <- th.cur
    sam$nu[i.sam] <- nu.cur
}

# update variance-covariance of the proposal
V = cov(cbind(sam$nu[1:N.test], sam$th[1:N.test]))

# resume sampling with updated variance-covariance matrix
for (i.sam in N.test + 1:N.sam) {
    pro <- mvrnorm(1, c(log(nu.cur), log(th.cur)), V)
    nu.pro <- exp(pro[1])
    th.pro <- exp(pro[2])

    p.cur <- n * nu.cur * log(th.cur) - n * lgamma(nu.cur) + nu.cur * (sum.log.x - 1) +
        3 * log(nu.cur) + log(th.cur) - th.cur * (2 + sum.x)
    p.pro <- n * nu.pro * log(th.pro) - n * lgamma(nu.pro) + nu.pro * (sum.log.x - 1) +
        3 * log(nu.pro) + log(th.pro) - th.pro * (2 + sum.x)

    accpt.prob <- exp(p.pro - p.cur)
    if (runif(1) < accpt.prob) {
        nu.cur <- nu.pro
        th.cur <- th.pro
    }

    sam$th[i.sam] <- th.cur
    sam$nu[i.sam] <- nu.cur
}

# generate summaries and plots
effectiveSize(sam$th)
```

```
##     var1
## 147.2699
```

```r
effectiveSize(sam$nu)
```

```
##     var1
## 247.0326
```

```r
mean(sam$nu)
```

```
## [1] 2.565554
```

```r
mean(sam$th)
```

```
## [1] 1.020709
```

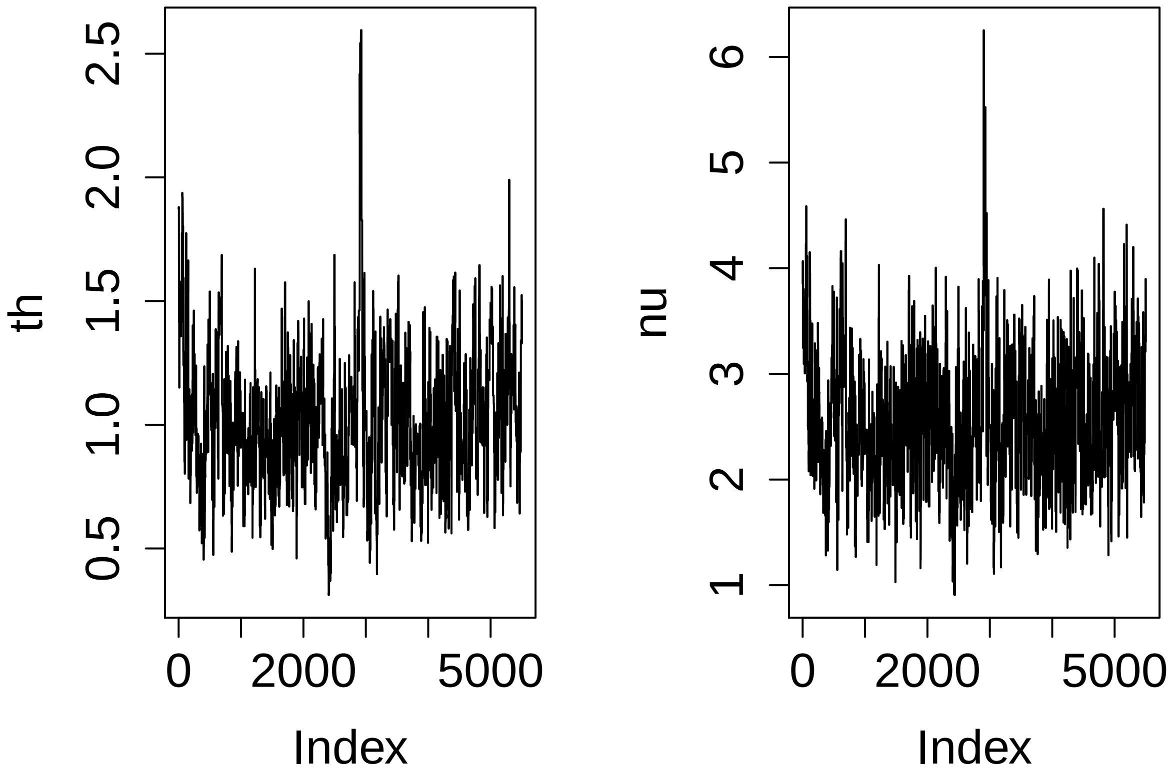
```r
quantile(sam$nu, probs = c(0.025, 0.975))
```

```
##     2.5%    97.5%
## 1.517385 3.895880
```

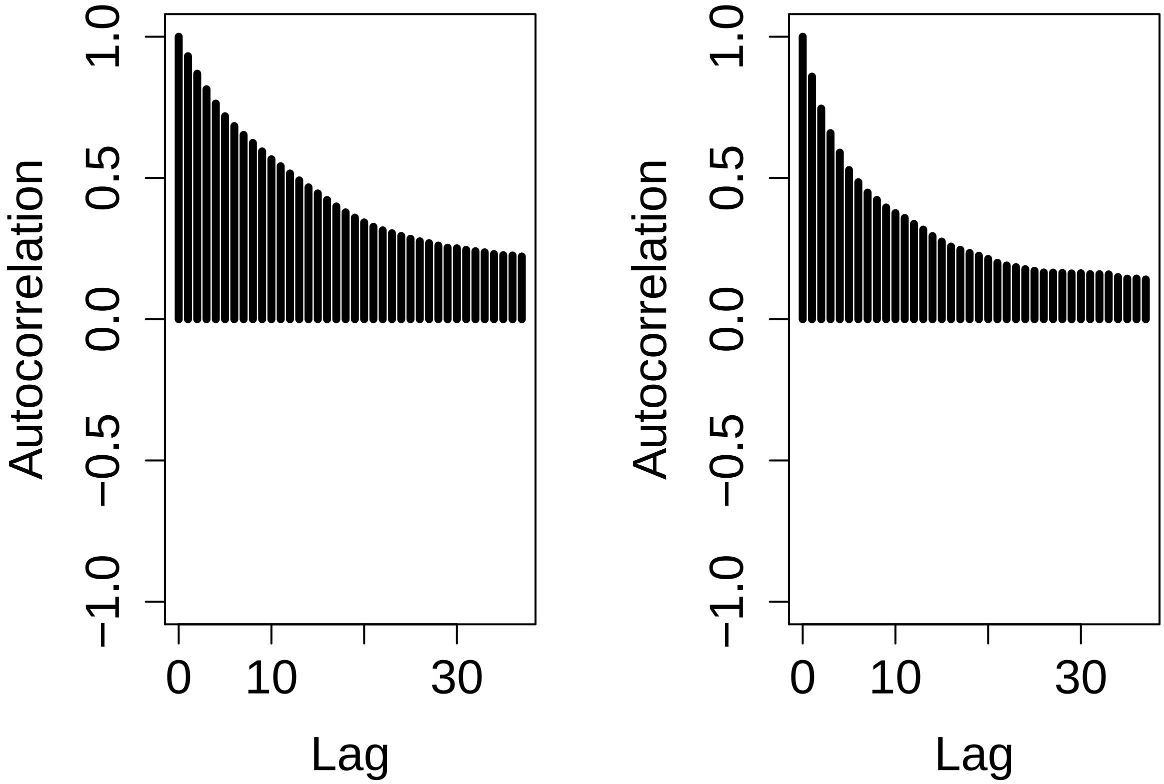
```r
quantile(sam$th, probs = c(0.025, 0.975))
```

```
##      2.5%     97.5%
## 0.5649966 1.5846521
```

```r
par(mfrow = c(1, 2), mar = c(4.5, 4.5, 2.1, 2.1))
plot(sam$th, col = 1, type = "l", main = "", cex.axis = 1.5, cex.lab = 1.5, ylab = "th")
plot(sam$nu, col = 1, type = "l", main = "", cex.axis = 1.5, cex.lab = 1.5, ylab = "nu")
```




```r
par(mfrow = c(1, 2), mar = c(4.5, 4.5, 2.1, 2.1))
autocorr.plot(sam$th, col = 1, lwd = 4, cex.axis = 1.5, cex.lab = 1.5, auto.layout = FALSE,
    main = "")
autocorr.plot(sam$nu, col = 1, lwd = 4, cex.axis = 1.5, cex.lab = 1.5, auto.layout = FALSE,
    main = "")
```



(c)

Let $t = \log(\theta)$ and $v = \log(\nu)$. Now, rewriting the posterior as:

$$\begin{aligned} p(\theta, \nu \mid \mathbf{x}) & \propto \exp\left\{(\nu - 1)\sum_{i=1}^{n}\log(x_i) + (a-1)\log(\nu) - b\nu - n\log\left(\Gamma(\nu)\right)\right\} \\ & \times \exp\left\{(\alpha + n\nu - 1)\log(\theta) - \theta\left(\beta + \sum_{i=1}^{n} x_i\right)\right\} \end{aligned}$$

we have that:

$$\begin{aligned} p(t, v \mid \mathbf{x}) & \propto \exp\left\{(e^v - 1)\sum_{i=1}^{n}\log(x_i) + av - be^v - n\log\left(\Gamma(e^v)\right)\right\} \\ & \times \exp\left\{(\alpha + ne^v)t - e^t\left(\beta + \sum_{i=1}^{n} x_i\right)\right\} \end{aligned}$$

Now let

$$h(t, v) = (e^v - 1)\sum_{i=1}^{n}\log(x_i) + av - be^v - n\log\left(\Gamma(e^v)\right) + (\alpha + ne^v)t - e^t\left(\beta + \sum_{i=1}^{n} x_i\right)$$

Finally, let $\hat{t}$ and $\hat{v}$ such that

$$h^{'}(\hat{t}, \hat{v}) = 0$$

then, we can use the Laplace approximation for $p(t, v \mid \mathbf{x})$ as follows:

$$p(t, v \mid \mathbf{x}) \approx C \exp\left\{-\frac{1}{2}((t, v) - (\hat{t}, \hat{v}))H(\hat{t}, \hat{v})((t, v) - (\hat{t}, \hat{v}))'\right\}$$

where $H(\hat{t}, \hat{v})$ is the Hessian matrix of h evaluated at $(\hat{t}, \hat{v})$. With this approximation, you can use an optimization function and ask for the Hessian obtained from it.

```
# Define the function to optimize
h = function(w) {
    r = (exp(w[2]) - 1) * sum.log.x + 3 * w[2] - exp(w[2])
    r = r - n * lgamma(exp(w[2])) + (2 + n * exp(w[2])) * w[1] - exp(w[1]) * (2 + sum.x)
    return(-r)
}

# Obtain the maximum and the Hessian at the maximum
lap = optim(c(0, 1), h, hessian = TRUE)

# Optimum
lap$par
```

```
## [1] 0.09685737 1.00165219
```

```
# Hessian
lap$hessian
```

```
##           [,1]      [,2]
## [1,]  70.06943 -68.06943
## [2,] -68.06943  85.06425
```

```
# initial values
th.cur <- exp(lap$par[1])
nu.cur <- exp(lap$par[2])

th.cur
```

```
## [1] 1.101703
```

```
nu.cur
```

```
## [1] 2.722777
```

```
# update variance-covariance of the proposal
V = diag(diag(solve(lap$hessian)))

# resume sampling with updated variance-covariance matrix
for (i.sam in N.test + 1:N.sam) {
    # propose (nu, theta) jointly from a Random Walk
    nu.pro <- exp(log(nu.cur) + rnorm(1, 0, sqrt(V[1, 1])))
    th.pro <- exp(log(th.cur) + rnorm(1, 0, sqrt(V[2, 2])))

    # evaluate the joint target density (on the log scale) at the current value
    p.cur <- (n * nu.cur + 2) * log(th.cur) - n * lgamma(nu.cur) + nu.cur * (sum.log.x -
        1) + 3 * log(nu.cur) - th.cur * (2 + sum.x)

    # evaluate the joint target density (on the log scale) at the proposed value
    p.pro <- (n * nu.pro + 2) * log(th.pro) - n * lgamma(nu.pro) + nu.pro * (sum.log.x -
        1) + 3 * log(nu.pro) - th.pro * (2 + sum.x)
```

```
    accpt.prob <- exp(p.pro - p.cur)
    if (runif(1) < accpt.prob) {
        nu.cur <- nu.pro
        th.cur <- th.pro
    }

    sam$th[i.sam] <- th.cur
    sam$nu[i.sam] <- nu.cur
}

# generate summaries and plots
effectiveSize(sam$th)
```

```
##     var1
## 171.0044
```

```
effectiveSize(sam$nu)
```

```
##    var1
## 191.267
```

```
mean(sam$nu)
```

```
## [1] 2.590977
```

```
mean(sam$th)
```

```
## [1] 1.052362
```

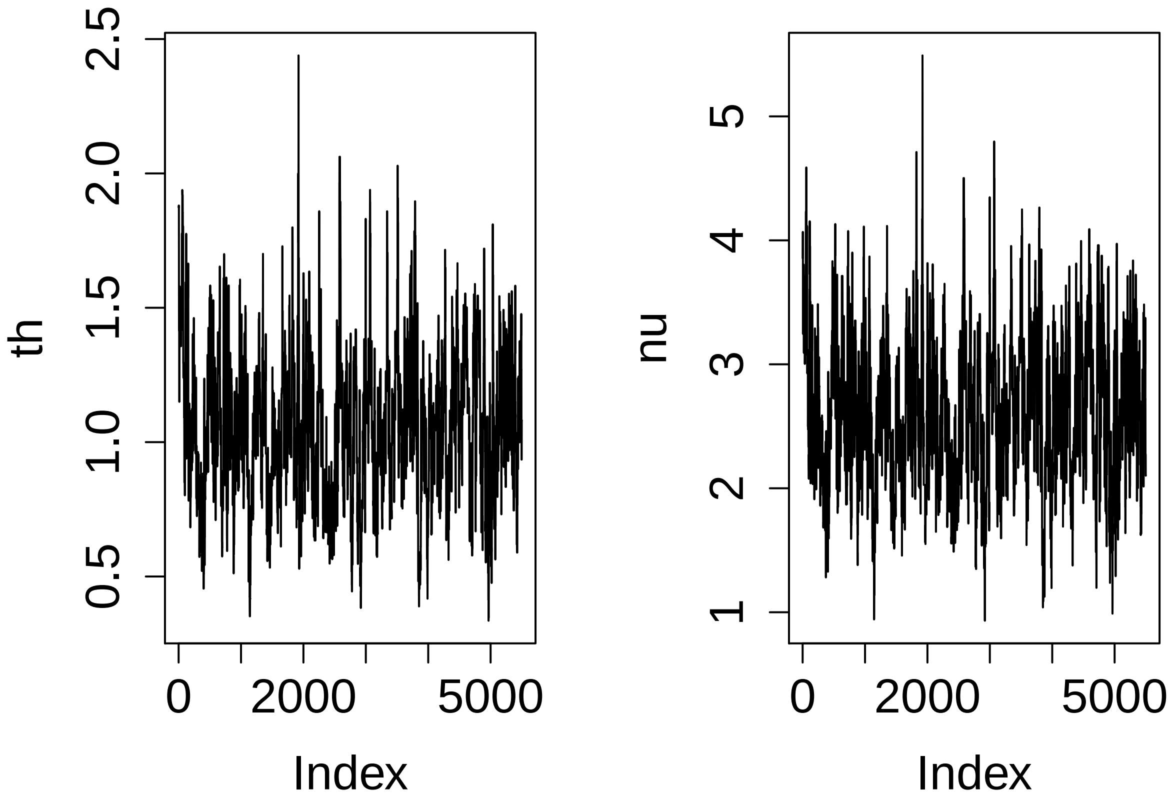
```
quantile(sam$nu, probs = c(0.025, 0.975))
```

```
##     2.5%    97.5%
## 1.537563 3.852109
```

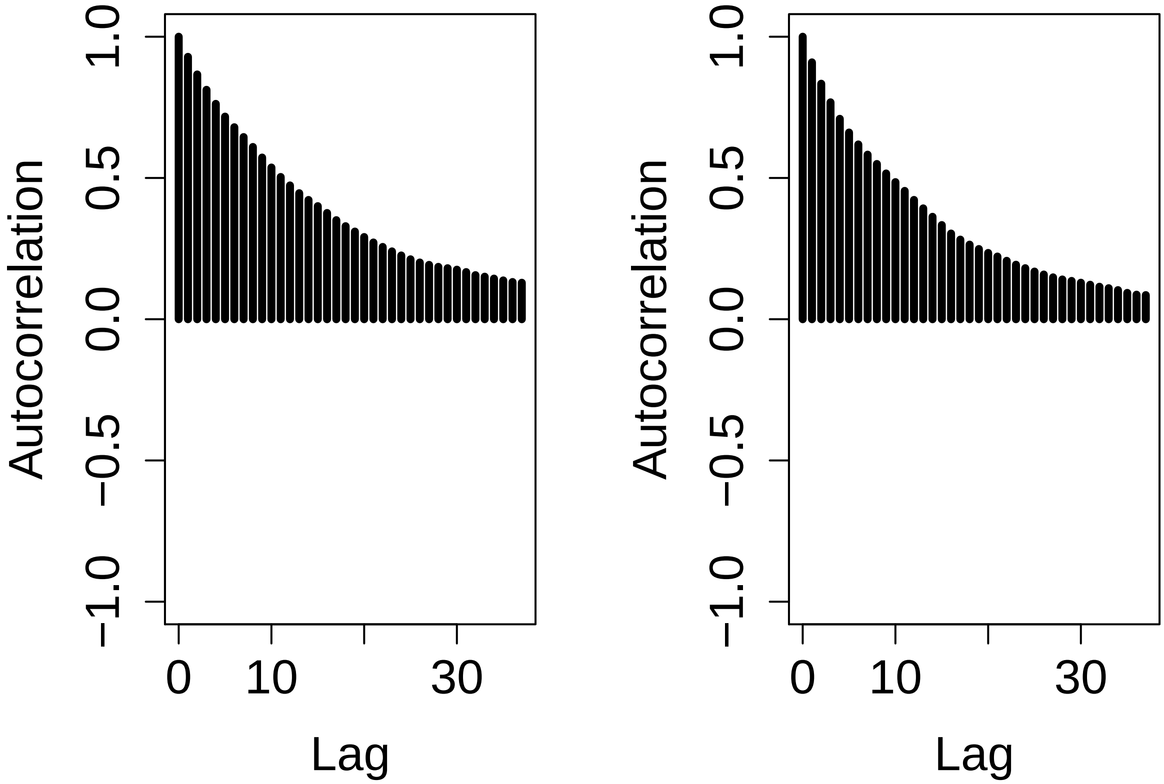
```
quantile(sam$th, probs = c(0.025, 0.975))
```

```
##      2.5%     97.5%
## 0.5667017 1.6607347
```

```
par(mfrow = c(1, 2), mar = c(4.5, 4.5, 2.1, 2.1))
plot(sam$th, col = 1, type = "l", main = "", cex.axis = 1.5, cex.lab = 1.5, ylab = "th")
plot(sam$nu, col = 1, type = "l", main = "", cex.axis = 1.5, cex.lab = 1.5, ylab = "nu")
```




```
par(mfrow = c(1, 2), mar = c(4.5, 4.5, 2.1, 2.1))
autocorr.plot(sam$th, col = 1, lwd = 4, cex.axis = 1.5, cex.lab = 1.5, auto.layout = FALSE,
    main = "")
autocorr.plot(sam$nu, col = 1, lwd = 4, cex.axis = 1.5, cex.lab = 1.5, auto.layout = FALSE,
    main = "")
```



(d) Same as before, but with Cauchy.

6. (Robert and Casella) Consider a random effects model,

$$y_{i,j} = \beta + u_i + \epsilon_{i,j}, \quad i = 1 : I, j = 1 : J,$$

where $u_i \sim N(0, \sigma^2)$ and $\epsilon_{i,j} \sim N(0, \tau^2)$. Assume a prior of the form

$$\pi(\beta, \sigma^2, \tau^2) \propto \frac{1}{\sigma^2 \tau^2}.$$

(a) Find the full conditional distributions:

i. $\pi(u_i | y, \beta, \tau^2, \sigma^2)$
ii. $\pi(\beta | y, u, \tau^2, \sigma^2)$
iii. $\pi(\sigma^2 | y, u, \beta, \tau^2)$
iv. $\pi(\tau^2 | y, u, \beta, \sigma^2)$

(b) Find $\pi(\beta, \tau^2, \sigma^2 | y)$ up to a proportionality constant.

(c) Find $\pi(\sigma^2, \tau^2 | y)$ up to a proportionality constant and show that this posterior is not integrable since, for $\tau \neq 0$, it behaves like σ^{-2} in a neighborhood of 0.

Note: This problem shows that even though the full conditional posteriors exist and the Gibbs sampling could be easily implemented, the joint posterior distribution does not exist. Users should be aware of the risks of using the Gibbs sampler in situations like this!

Solution:

(a) The joint posterior for this model satisfies

$$\pi(u, \beta, \tau^2, \sigma^2 | y) \propto \left(\tau^2\right)^{-\left(\frac{IJ}{2}+1\right)} \left(\sigma^2\right)^{-\left(\frac{I}{2}+1\right)} \exp\left\{ -\frac{1}{2\tau^2} \sum_{i,j} (y_{i,j} - (\beta + u_i))^2 - \frac{1}{2\sigma^2} \sum_i u_i^2 \right\}.$$

Thus, the full conditionals are given as follows:

i. $\pi(u_i|y,\beta,\tau^2,\sigma^2) \propto \exp\left\{-\frac{1}{2}\left[u_i^2\left(\frac{J}{\tau^2}+\frac{1}{\sigma^2}\right)-2u_i\left(\frac{\sum_j (y_{i,j}-\beta)}{\tau^2}\right)\right]\right\}$. That is,

$$u_i \mid y,\tau^2,\sigma^2,\beta \overset{ind.}{\sim} \mathcal{N}\left(\left(\frac{J}{\tau^2}+\frac{1}{\sigma^2}\right)^{-1}\left(\frac{\sum_j (y_{ij}-\beta)}{\tau^2}\right), \left(\frac{J}{\tau^2}+\frac{1}{\sigma^2}\right)^{-1}\right)$$

ii. $\pi(\beta|y,u,\tau^2,\sigma^2) \propto \exp\left\{-\frac{1}{2}\left[\beta^2\frac{IJ}{\tau^2}-2\beta\left(\frac{\sum_{i,j}(y_{i,j}-u_i)}{\tau^2}\right)\right]\right\}$. That is,

$$\beta \mid y,\tau^2,\sigma^2,u \sim \mathcal{N}\left(\left(\frac{IJ}{\tau^2}\right)^{-1}\left(\frac{\sum_{i,j}(y_{ij}-u_i)}{\tau^2}\right), \left(\frac{IJ}{\tau^2}\right)^{-1}\right).$$

iii. $\pi(\sigma^2|y,u,\beta,\tau^2) \propto \left(\sigma^2\right)^{-\left(\frac{I}{2}+1\right)}\exp\left\{\frac{-\frac{1}{2}\sum_i u_i^2}{\sigma^2}\right\}$. That is,

$$\sigma^2|y,u,\beta,\tau^2 \sim \mathcal{IG}\left(\frac{I}{2},\frac{1}{2}\sum_i u_i^2\right)$$

iv. $\pi(\tau^2|y,u,\beta,\sigma^2) \propto \left(\tau^2\right)^{-\left(\frac{IJ}{2}+1\right)}\exp\left\{\frac{-\frac{1}{2}\sum_{i,j}(y_{i,j}-(\beta+u_i))^2}{\tau^2}\right\}$. That is,

$$\tau^2|y,u,\beta,\tau^2 \sim \mathcal{IG}\left(\frac{IJ}{2},\frac{1}{2}\sum_{i,j}(y_{i,j}-(\beta+u_i))^2\right)$$

(b) $\pi(\beta,\tau^2,\sigma^2|y) = \int\prod_i \pi(u_i,\beta,\tau^2,\sigma^2|y)du_i$. Therefore,

$$\pi(\beta,\tau^2,\sigma^2|y) \propto \left(\tau^2\right)^{-\left(\frac{I(J-1)}{2}+1\right)}\left(\sigma^2\right)^{-1}\left(J\sigma^2+\tau^2\right)^{\frac{I}{2}}$$
$$\exp\left\{-\frac{1}{2\tau^2}\sum_{i,j}(y_{i,j}-\beta)^2\right\}\exp\left\{\frac{\sigma^2}{2\tau^2(J\sigma^2+\tau^2)}\sum_i\left(\sum_j (y_{i,j}-\beta)\right)^2\right\}$$

(c) $\pi(\tau^2,\sigma^2|y) = \int \pi(\beta,\tau^2,\sigma^2|y)d\beta$. Therefore,

$$\pi(\tau^2,\sigma^2|y) \propto \left(\tau^2\right)^{-\left(\frac{I(J-1)}{2}+1\right)}\left(\sigma^2\right)^{-1}\left(J\sigma^2+\tau^2\right)^{\frac{I+1}{2}}\exp\left\{-\frac{1}{2\tau^2}\sum_{i,j}y_{i,j}^2\right\}$$
$$\exp\left\{\frac{\sigma^2}{2\tau^2(J\sigma^2+\tau^2)}\sum_i\left(\sum_j y_{i,j}\right)^2\right\}\exp\left\{\frac{1}{2IJ(J\sigma^2+\tau^2)}\left(\sum_{i,j}y_{i,j}\right)^2\right\}$$

and notice that, for a fixed τ, as $\sigma^2 \to 0$ $\pi(\tau^2,\sigma^2|y) \propto \left(\sigma^2\right)^{-1}$ which is not integrable w.r.t. σ^2.

7. (Carlin, Gelfand and Smith, 1992) Let $y_1,\ldots,y_n$ be a sample from a Poisson distribution for which there is a suspicion of a change point m along the observation process where the means change, $m = 1,\ldots,n$. Given m, $y_i \sim \mathcal{P}oi(\theta)$, for $i = 1,\ldots,m$ and $y_i \sim \mathcal{P}oi(\phi)$, for $i = m+1,\ldots,n$. The model is completed with independent prior distributions $\lambda \sim \mathcal{G}a(\alpha,\beta)$, $\phi \sim \mathcal{G}a(\gamma,\delta)$ and m uniformly distributed over $\{1,\ldots,n\}$ where α,β,γ and δ are known constants. Implement a Gibbs sampling algorithm to obtain samples from the joint posterior distribution. Run the Gibbs sampler to apply this model to the data *mining.r* which consists of counts of coal mining disasters in Great Britain by year from 1851 to 1962.

Solution:

The joint posterior for this model is such that

$$p(\theta, \phi, m \mid \mathbf{y}) \propto \theta^{\alpha+\sum_{i=1}^{m} y_i - 1} \exp\left\{-\left(\beta+m\right)\theta\right\} \phi^{\gamma+\sum_{i=m+1}^{n} y_i - 1} \exp\left\{-\left(\delta+n-m\right)\phi\right\}$$

So that the full conditionals are given by

- $\theta \mid m, \mathbf{y} \sim \mathcal{G}a(\alpha + \sum_{i=1}^{m} y_i, \beta + m)$
- $\phi \mid m, \mathbf{y} \sim \mathcal{G}a(\gamma + \sum_{i=m+1}^{n} y_i, \delta + n - m)$

and

- $p(m \mid \theta, \phi, \mathbf{y}) \propto \theta^{\alpha+\sum_{i=1}^{m} y_i - 1} \exp\{-(\beta+m)\theta\}\phi^{\gamma+\sum_{i=m+1}^{n} y_i - 1} \exp\{-(\delta+n-m)\phi\}$

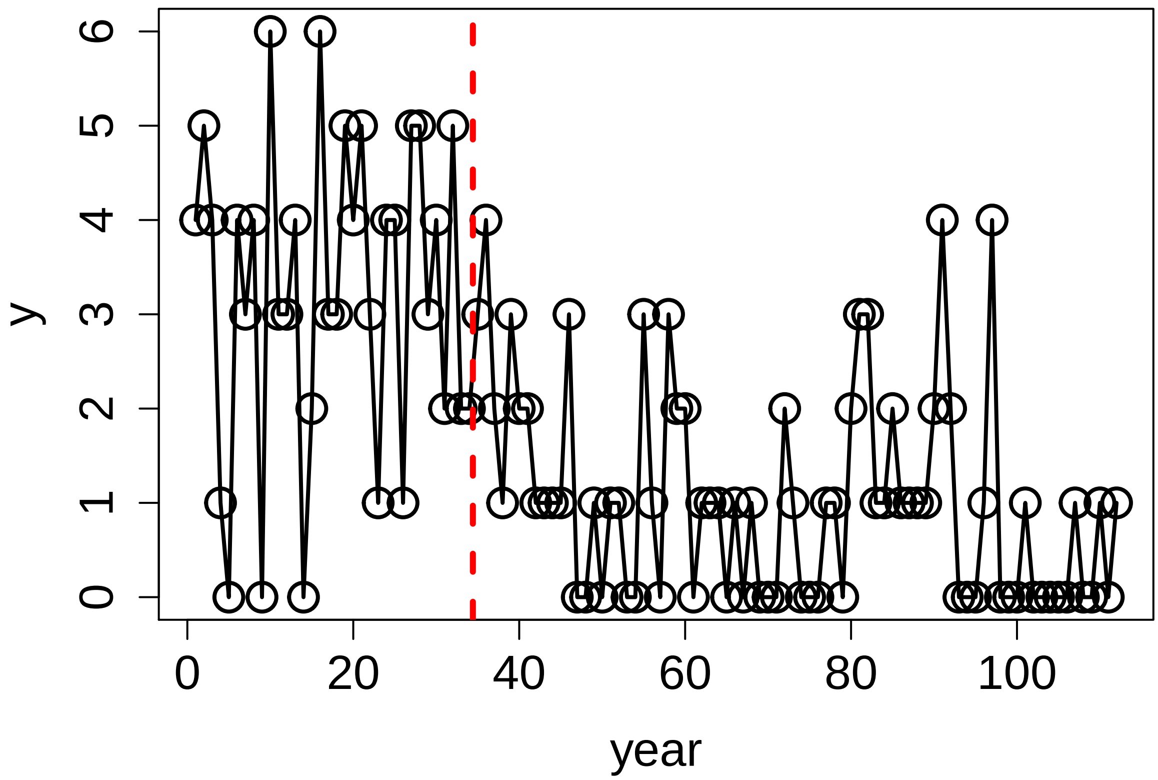
From these is possible to directly derive a Gibbs sampler to sample from the joint posterior distribution. However, because of autocorrelation issues, this sampler will tend to be inefficient. Instead, it is possible to collapse over (integrate out) θ and ϕ to get that

- $p(m \mid \mathbf{y}) = \int p(m \mid \theta, \phi, \mathbf{y}) d_\theta d_\phi \propto \frac{\Gamma\left(\alpha+\sum_{i=1}^{m} y_i\right)}{(\beta+m)^{\alpha+\sum_{i=1}^{m} y_i}} \frac{\Gamma\left(\gamma+\sum_{i=m+1}^{n} y_i\right)}{(\delta+n-m)^{\gamma+\sum_{i=m+1}^{n} y_i}}$

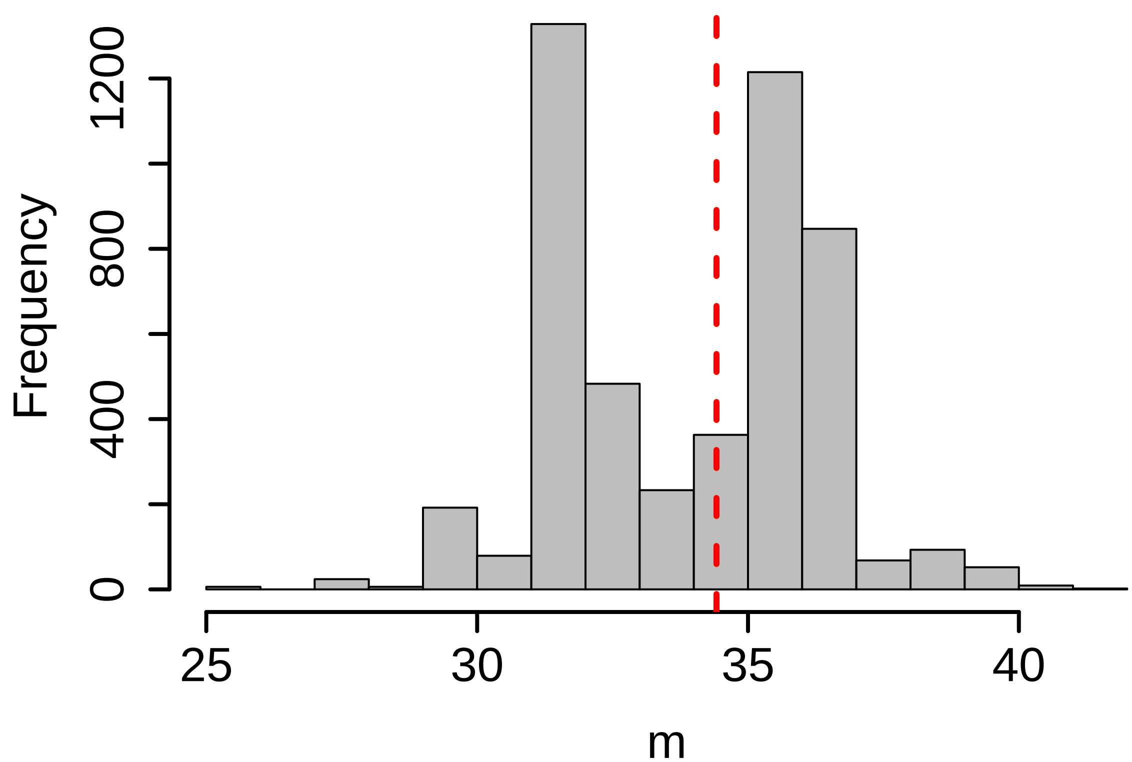
which can be used in a Metropolis-within-Gibbs algorithm as follws:

```r
# input data
y <-c(4,5,4,1,0,4,3,4,0,6,3,3,4,0,2,6,3,3,5,4,5,3,1,4,4,1,5,5,3,4,2,5,2,2,3,4,2,1,3,2,2,1,
1,1,1,3,0,0,1,0,1,1,0,0,3,1,0,3,2,2,0,1,1,1,0,1,0,1,0,0,0,2,1,0,0,0,1,1,0,2,3,3,1,1,2,1,1,
1,1,2,4,2,0,0,0,1,4,0,0,0,1,0,0,0,0,0,1,0,0,1,0,1)
n <- length(y)

# specify hyperparameters. In this case using the data.
alpha <- 3
beta <- alpha/mean(y[1:40])
gam <- 3
delta <- gam/mean(y[-(1:40)])

# set up MCMC variables
N.sam <- 5000
N.burn <- 500
save.sam <- NULL
save.sam$m <- rep(NA, N.sam)
save.sam$th <- rep(NA, N.sam)
save.sam$phi <- rep(NA, N.sam)

# initialize chains
th.cur <- rgamma(1, alpha, beta)
phi.cur <- rgamma(1, gam, delta)
m.cur <- 40

# sampling
for(i.sam in 1:N.sam)
{
    # update theta from its full conditional
    th.cur <- rgamma(1, sum(y[1:m.cur]) + alpha, m.cur + beta)
```

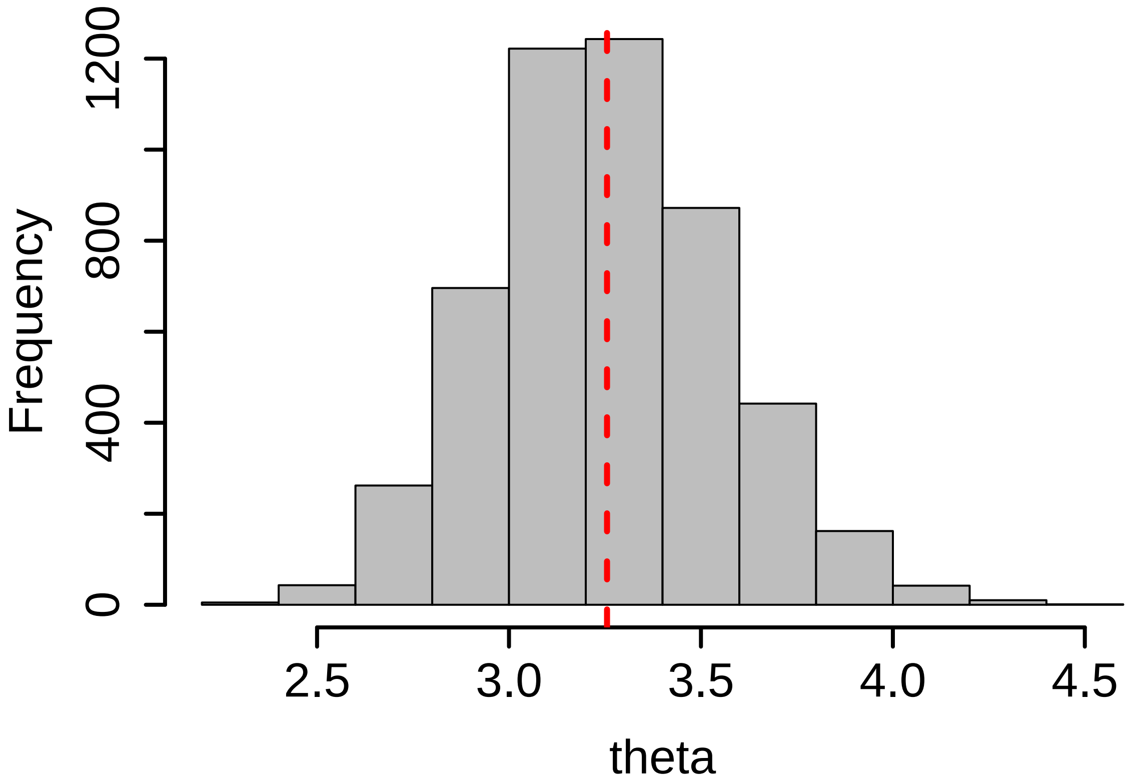
```r
    # update phi from its full conditional
    phi.cur <- rgamma(1, sum(y[-(1:m.cur)]) + gam, (n-m.cur + delta))

    # get a proposed value for m from a uniform proposal in {1,2,...,n}
    m.pro <- sample((1:n), 1, FALSE)

    # evaluate the pdf (log-scale) at the current value of m
    p.cur <- lgamma(sum(y[1:m.cur]) + alpha) - (sum(y[1:m.cur]) + alpha)*log(m.cur + beta)
    +lgamma(sum(y[-(1:m.cur)]) + gam) - (sum(y[-(1:m.cur)]) + gam)*log(n-m.cur + delta)

    # evaluate the pdf (log-scale) at the proposed value of m
    p.pro <- lgamma(sum(y[1:m.pro]) + alpha) - (sum(y[1:m.pro]) + alpha)*log(m.pro + beta)
    +lgamma(sum(y[-(1:m.pro)]) + gam) - (sum(y[-(1:m.pro)]) + gam)*log(n-m.pro + delta)

    # calculate acceptance probability and accept/reject acordingly
    accpt.prob <- exp(p.pro - p.cur)
    if(runif(1) < accpt.prob)
    {
        m.cur <- m.pro
    }

    # save the current draws
    save.sam$th[i.sam] <- th.cur
    save.sam$phi[i.sam] <- phi.cur
    save.sam$m[i.sam] <- m.cur
}

# plot evolution of y along with posterior mean estimate of m
par(mar=c(4.5, 4.5, 2.1, 2.1))
plot((1:n), y, cex=2, lwd=2, type="o", main="", cex.axis=1.5, cex.lab=1.5, xlab="year", ylab="y")
abline(v=mean(save.sam$m), lty=2, lwd=3, col=2)
```



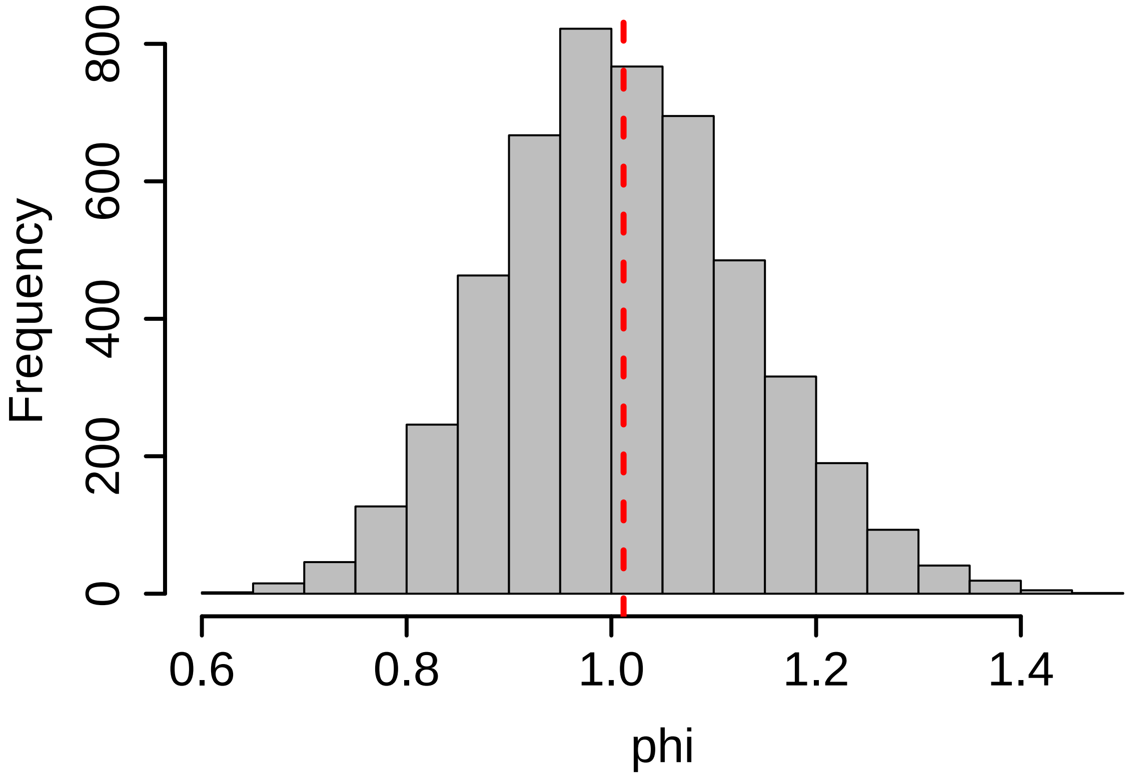

```r
# plot histogram of m, red dotted vertical line represents posterior mean
par(mar=c(4.5, 4.5, 2.1, 2.1))
hist(save.sam$m, col=8, lwd=2, , main="", cex.axis=1.5, cex.lab=1.5, xlab="m")
abline(v=mean(save.sam$m), lty=2, lwd=3, col=2)
```



```
# plot histogram of theta, red dotted vertical line represents posterior mean
par(mar=c(4.5, 4.5, 2.1, 2.1))
hist(save.sam$th, col=8, lwd=2, , main="", cex.axis=1.5, cex.lab=1.5, xlab="theta")
abline(v=mean(save.sam$th), lty=2, lwd=3, col=2)
```




```
# plot histogram of phi, red dotted vertical line represents posterior mean
par(mar=c(4.5, 4.5, 2.1, 2.1))
hist(save.sam$phi, col=8, lwd=2, , main="", cex.axis=1.5, cex.lab=1.5, xlab="phi")
abline(v=mean(save.sam$phi), lty=2, lwd=3, col=2)
```



8. Souza (1999) considers a number of hierarchical models to describe the nutritional pattern of pregnant women. One of the models adopted was a hierarchical regression model where

$$\begin{aligned} y_{i,j} &\sim \mathcal{N}(\alpha_i + \beta_i t_{i,j}, \sigma^2), \\ (\alpha_i, \beta_i)' | \alpha, \beta &\sim \mathcal{N}_2((\alpha, \beta)', diag(\tau_\alpha^{-1}, \tau_\beta^{-1})), \\ (\alpha, \beta)' &\sim \mathcal{N}_2((0,0)', diag(P_\alpha^{-1}, P_\beta^{-1}), \end{aligned}$$

prior independent scale parameters σ^{-2}, τ_α and $\tau_\beta \sim \mathcal{G}a(a, b)$, and $y_{i,j}$ and $t_{i,j}$ are the jth weight measurement and visit time of the ith woman with $j = 1 : n_i$ and $i = 1 : I$ for $I = 68$ pregnant women. Here $n = \sum_{i=1}^{I} n_i = 427$, $P_\alpha = P_\beta = 1/1000$ and $a = b = 0.001$. Find the full conditional distributions of $\alpha, \beta, \tau_\alpha, \tau_\beta$, σ^{-2}, α_i, β_i, and (α_i, β_i).

Solution:

Let θ denote a vector of parameters, $\theta = ((\alpha_i, \beta_i)_{i=1}^{I}, \alpha, \beta, \sigma^2, \tau_\alpha, \tau_\beta)$. The joint posterior distribution of θ satisfies

$$\begin{aligned} p(\theta \mid \mathbf{y}) \propto \left(\sigma^{-2}\right)^{\frac{n}{2}} \exp\left\{-\frac{1}{2\sigma^2}\sum_{i=1}^{I}\sum_{j=1}^{n_i}(y_{i,j} - (\alpha_i + \beta_i t_{i,j}))^2\right\} (\tau_\alpha)^{I/2}(\tau_\beta)^{I/2} \exp\left\{-\frac{1}{2}\left[P_\alpha \alpha^2 + P_\beta \beta^2\right]\right\} \\ \times \exp\left\{-\frac{1}{2}\left[\tau_\alpha \sum_{i=1}^{I}(\alpha_i - \alpha)^2 + \tau_\beta \sum_{i=1}^{I}(\beta_i - \beta)^2\right]\right\} \left(\sigma^{-2}\tau_\alpha\tau_\beta\right)^{a-1} \exp\left\{-b(\sigma^{-2} + \tau_\alpha + \tau_\beta)\right\}, \end{aligned}$$

so the full conditionals are given by

- $\alpha \mid \mathbf{y}, \theta_{-\alpha} \sim \mathcal{N}\left((I\tau_\alpha + P_\alpha)^{-1}\left(\tau_\alpha \sum_{i=1}^{I} \alpha_i\right), (I\tau_\alpha + P_\alpha)^{-1}\right)$

- $\beta \mid \mathbf{y}, \theta_{-\beta} \sim \mathcal{N}\left((I\tau_\beta + P_\beta)^{-1}\left(\tau_\beta \sum_{i=1}^{I} \beta_i\right), (I\tau_\beta + P_\beta)^{-1}\right)$
- $\tau_\alpha \mid \mathbf{y}, \theta_{-\tau_\alpha} \sim \mathcal{G}a\left(a + \frac{I}{2}, b + \frac{1}{2}\sum_{i=1}^{I}(\alpha_i - \alpha)^2)\right)$
- $\tau_\beta \mid \mathbf{y}, \theta_{-\tau_\beta} \sim \mathcal{G}a\left(a + \frac{I}{2}, b + \frac{1}{2}\sum_{i=1}^{I}(\beta_i - \beta)^2)\right)$
- $\sigma^{-2} \mid \mathbf{y}, \theta_{-\sigma^{-2}} \sim \mathcal{G}a\left(a + \frac{n}{2}, b + \frac{1}{2}\sum_{i=1}^{I}\sum_{j=1}^{n_i}(y_{i,j} - (\alpha_i + \beta_i t_{i,j}))^2\right)$
- $\alpha_i \mid \mathbf{y}, \theta_{-\alpha_i} \overset{ind.}{\sim} \mathcal{N}\left((\tau_\alpha + \sigma^{-2} n_i)^{-1}(\alpha\tau_\alpha + \sigma^{-2}\sum_{j=1}^{n_i}(y_{i,j} - \beta_i t_{i,j})), (\tau_\alpha + \sigma^{-2} n_i)^{-1}\right)$
- $\beta_i \mid \mathbf{y}, \theta_{-\beta_i} \overset{ind.}{\sim} \mathcal{N}\left((\tau_\beta + \sigma^{-2}\sum_{j=1}^{n_i} t_{i,j}^2)^{-1}(\beta\tau_\beta + \sigma^{-2}\sum_{j=1}^{n_i}(y_{i,j} - \alpha_i)), (\tau_\beta + \sigma^{-2}\sum_{j=1}^{n_i} t_{i,j}^2)^{-1}\right)$
- Let $\boldsymbol{\xi}_i = (\mathbf{1}_{n_i}, \mathbf{t}_i)$ and $T^{-1} = diag(\tau_\alpha, \tau_\beta)$, then $(\alpha_i, \beta_i) \overset{ind.}{\sim} \mathcal{N}_2(\boldsymbol{\mu}_i, \Sigma_i)$ where

$$\boldsymbol{\mu}_i = \Sigma_i\left(\sigma^{-2}\boldsymbol{\xi}_i'\mathbf{y}_i + T^{-1}(\alpha, \beta)'\right) \quad \text{and} \quad \Sigma_i = \left(\sigma^{-2}\boldsymbol{\xi}_i'\boldsymbol{\xi}_i + T^{-1}\right)^{-1}$$