STAT 35920: Homework 5

Robert Winter

Table of Contents

1 Exercise 1

Two genes are believed to co-express in human bodies. Measurements of gene expression for both genes are standardized to follow a $\mathcal{N}(0,1)$ distribution. We want to investigate the correlation ρ between the two genes and obtained the measurements $(X_1, Y_1), \ldots, (X_n, Y_n)$ of both genes in $n = 13$ individuals. Here X_i and Y_i denote the expression of Gene One and Two, respectively. The data are

x_i	0.92	0.42	3.62	0.89	-0.69	0.45	-0.11	-0.14	-0.47	1.09	-0.34	0.62	0.27
y_i	0.26	1.65	2.10	0.62	-1.16	1.29	-0.82	-0.36	-0.29	0.86	0.19	1.25	0.33

1.1 Part (a)

Derive the likelihood function assuming (X_i, Y_i) follows a bivariate normal distribution with marginal distribution $\mathcal{N}(0,1)$ and correlation ρ.

Since $X \sim \mathcal{N}(0,1)$, $Y \sim \mathcal{N}(0,1)$, $\text{Corr}(X,Y) = \rho$, and we know the joint distribution of (X,Y) is bivariate normal, the joint distribution of (X,Y) is

$$(X_i, Y_i) \sim \mathcal{N}_2\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix} \right)$$

Thus, the joint density of (X,Y) is given by

$$\begin{aligned} f(x,y) &= \frac{1}{2\pi\sigma_X\sigma_Y\sqrt{1-\rho^2}} \exp\left\{ -\frac{1}{2(1-\rho^2)} \left[\left(\frac{x-\mu_X}{\sigma_X}\right)^2 - 2\rho\left(\frac{x-\mu_X}{\sigma_X}\right)\left(\frac{y-\mu_Y}{\sigma_Y}\right) + \left(\frac{y-\mu_Y}{\sigma_Y}\right)^2 \right] \right\} \\ &= \frac{1}{2\pi(1)(1)\sqrt{1-\rho^2}} \exp\left\{ -\frac{1}{2(1-\rho^2)} \left[\left(\frac{x-0}{1}\right)^2 - 2\rho\left(\frac{x-0}{1}\right)\left(\frac{y-0}{1}\right) + \left(\frac{y-0}{1}\right)^2 \right] \right\} \\ &= \frac{1}{2\pi\sqrt{1-\rho^2}} \exp\left\{ -\frac{1}{2(1-\rho^2)} [x^2 - 2\rho xy + y^2] \right\}. \end{aligned}$$

So, the likelihood function is

$$\begin{aligned} \mathcal{L}(\rho|\mathbf{x},\mathbf{y}) &= \prod_{i=1}^{13} f(x_i, y_i) \\ &= \prod_{i=1}^{13} \frac{1}{2\pi\sqrt{1-\rho^2}} \exp\left\{ -\frac{1}{2(1-\rho^2)} [x_i^2 - 2\rho x_i y_i + y_i^2] \right\} \\ &= \frac{1}{(2\pi)^{13}(1-\rho^2)^{13/2}} \exp\left\{ -\frac{1}{2(1-\rho^2)} \sum_{i=1}^{13} [x_i^2 - 2\rho x_i y_i + y_i^2] \right\} \\ &= \frac{1}{(2\pi)^{13}(1-\rho^2)^{13/2}} \exp\left\{ -\frac{1}{2(1-\rho^2)} \left[\sum_{i=1}^{13} x_i^2 - 2\rho \sum_{i=1}^{13} x_i y_i + \sum_{i=1}^{13} y_i^2 \right] \right\} \end{aligned}$$

For the given data, $\sum_{i=1}^{13} x_i^2 = 17.6115$, $\sum_{i=1}^{13} x_i y_i = 12.4807$, and $\sum_{i=1}^{13} y_i^2 = 13.9274$.

```
sum(genes$X^2); sum(genes$X * genes$Y); sum(genes$Y^2)
```

```
[1] 17.6115
```

```
[1] 12.4807
```

```
[1] 13.9274
```

Thus, continuing,

$$\begin{aligned}\mathcal{L}(\rho|\mathbf{x},\mathbf{y}) &= \frac{1}{(2\pi)^{13}(1-\rho^2)^{13/2}}\exp\Big\{-\frac{1}{2(1-\rho^2)}[17.6115-2\rho(12.4807)+13.9274]\Big\}\\ &= \frac{1}{(2\pi)^{13}(1-\rho^2)^{13/2}}\exp\Big\{-\frac{1}{2(1-\rho^2)}[31.5389-24.9614\rho]\Big\}.\end{aligned}$$

1.2 Part (b)

Suppose that ρ follows a $\mathcal{U}[0,1]$ prior. Consider a Metropolis-Hastings sampling algorithm for ρ. Suppose the proposal density of ρ is

i. **Given the current value $\rho^{(s)}$, sample $\rho^* \sim \mathcal{U}[\rho^{(s)}-0.2, \rho^{(s)}+0.2]$.**

ii. **If the sampled $\rho^* < 0$, then set $\rho^* = |\rho^*|$.**

iii. **If the sampled $\rho^* > 1$, then set $\rho^* = 2-\rho^*$.**

Argue that this is a symmetric proposal density.

The proposal density is:

$$\begin{aligned}q(\rho^*|\rho^{(s)}) &= \frac{1}{(\rho^{(s)}+0.2)-(\rho^{(s)}-0.2)}\mathbb{I}_{\{\rho^{(s)}-0.2\le\rho^*\le\rho^{(s)}+0.2\}}\\ &= \frac{1}{\rho^{(s)}+0.2-\rho^{(s)}+0.2}\mathbb{I}_{\{\rho^{(s)}-0.2\le\rho^*\le\rho^{(s)}+0.2\}}\\ &= \frac{1}{0.4}\mathbb{I}_{\{\rho^{(s)}-0.2\le\rho^*\le\rho^{(s)}+0.2\}}\\ &= 2.5\mathbb{I}_{\{\rho^{(s)}-0.2\le\rho^*\le\rho^{(s)}+0.2\}}\end{aligned}$$

So,

$$\begin{aligned}q(\rho^{(s)}|\rho^*) &= 2.5\mathbb{I}_{\{\rho^*-0.2\le\rho^{(s)}\le\rho^*+0.2\}}\\ &= 2.5\mathbb{I}_{\{\rho^*-0.2\le\rho^{(s)}\}}\mathbb{I}_{\{\rho^{(s)}\le\rho^*+0.2\}}\\ &= 2.5\mathbb{I}_{\{\rho^*\le\rho^{(s)}+0.2\}}\mathbb{I}_{\{\rho^{(s)}-0.2\le\rho^*\}}\\ &= 2.5\mathbb{I}_{\{\rho^{(s)}-0.2\le\rho^*\le\rho^{(s)}+0.2\}}\\ &= q(\rho^*|\rho^{(s)}).\end{aligned}$$

Thus, $q(\bullet|\bullet)$ is symmetric.

1.3 Part (c)

Sample from the posterior distribution of ρ using the M-H algorithm with the above proposal density. Make trace plots and check autocorrelations. See if thinning will reduce autocorrelation. Summarize the posterior distribution of ρ using the MCMC samples.

We sample from the posterior distribution of ρ using the Metropolis-Hastings algorithm below, using an initial value of $\rho^{(s)} = 0.5$ and 10,000 iterations of the algorithm.

```
nsim = 10000 # iterations of algo
rho1 = 0.5 # initial value
acceptances = 0 # counter for acceptance rate

rho_values = matrix(0, nrow = nsim) # store rho values for each
 ↪  iteration of MH
rho_values[1] = rho1

set.seed(41) # favorite number

# MH algorithm
for(i in 2:nsim){
  # Step 1: Draw rhostar from proposal density
  rhostar = runif(1, min = rho_values[i-1] - 0.2,
                     max = rho_values[i-1] + 0.2)
  if(rhostar<0){
    rhostar = abs(rhostar)
  }
  else if(rhostar>1){
    rhostar = 2 - rhostar
  }

  # Step 2: Acceptance/Odds ratio
  ### Logs of priors
  prior_rhoS = dunif(rho_values[i-1], log=T)
  prior_rhostar = dunif(rhostar, log=T)

  ### Logs of likelihoods
  lik_rhoS = log(1/((2*pi)^13*(1-rho_values[i-1]^2)^(13/2))*

                 ↪  exp(-1/(2*(1-rho_values[i-1]^2))*(31.5389-24.9614*rho_values[i-1])))
  lik_rhostar = log(1/((2*pi)^13*(1-rhostar^2)^(13/2))*
                  exp(-1/(2*(1-rhostar^2))*(31.5389-24.9614*rhostar)))
```

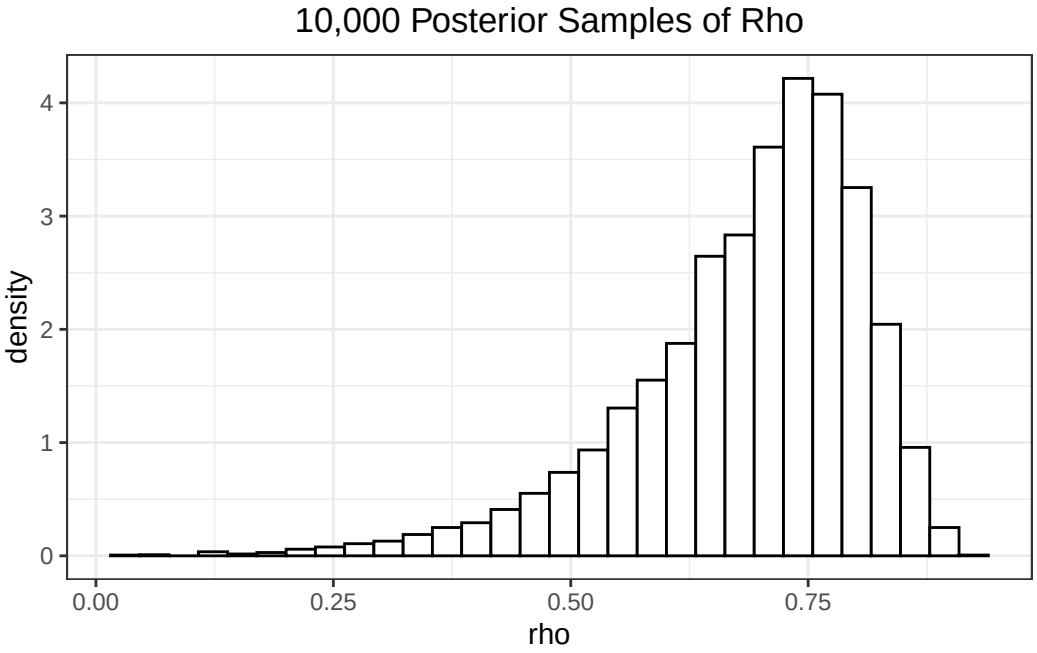
```
  ### Acceptance ratio
  log_accept_ratio = (lik_rhostar + prior_rhostar) - (lik_rhoS +
↪   prior_rhoS)
  accept_ratio = exp(log_accept_ratio)

  # Steps 3/4: Accept/Reject
  if(accept_ratio >= 1){
    # auto-accept if r >= 1
    rho_values[i] = rhostar
    acceptances = acceptances + 1
  }
  else if(accept_ratio < 1){
    # Draw random number from [0,1]
    draw = runif(1, 0, 1)
    if(draw <= accept_ratio){
      # Accept
      rho_values[i] = rhostar
      acceptances = acceptances + 1
    }
    else{
      # Reject
      rho_values[i] = rho_values[i-1]
    }
  }
}
```

Our algorithm's acceptance rate was 63.27%, which falls squarely in the desirable $[20\%, 80\%]$ range for convergence.

```
# Acceptance rate
acceptances/nsim # ~63%
```

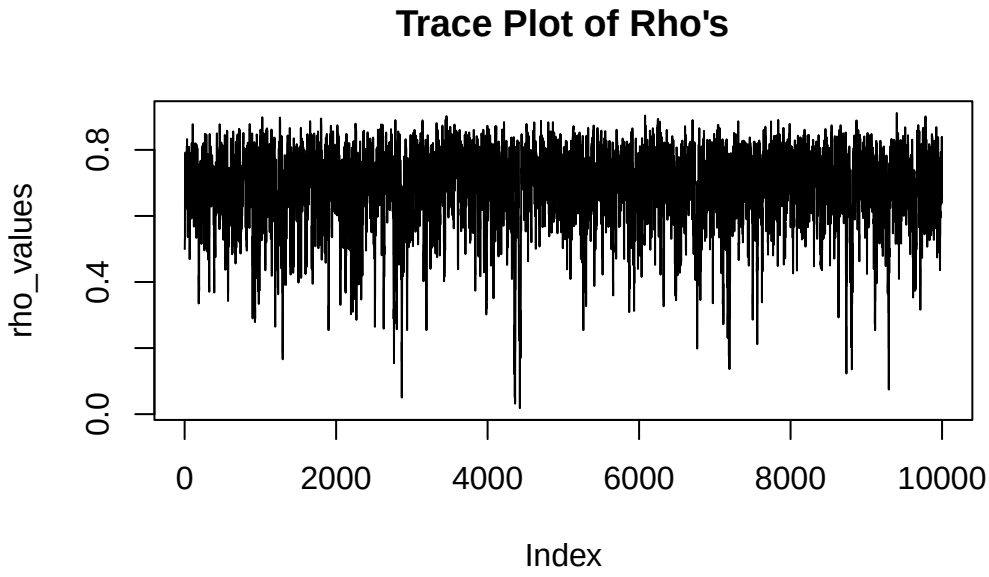
```
[1] 0.6327
```

We plot the posterior density of ρ based on our Metropolis-Hastings algorithm below:



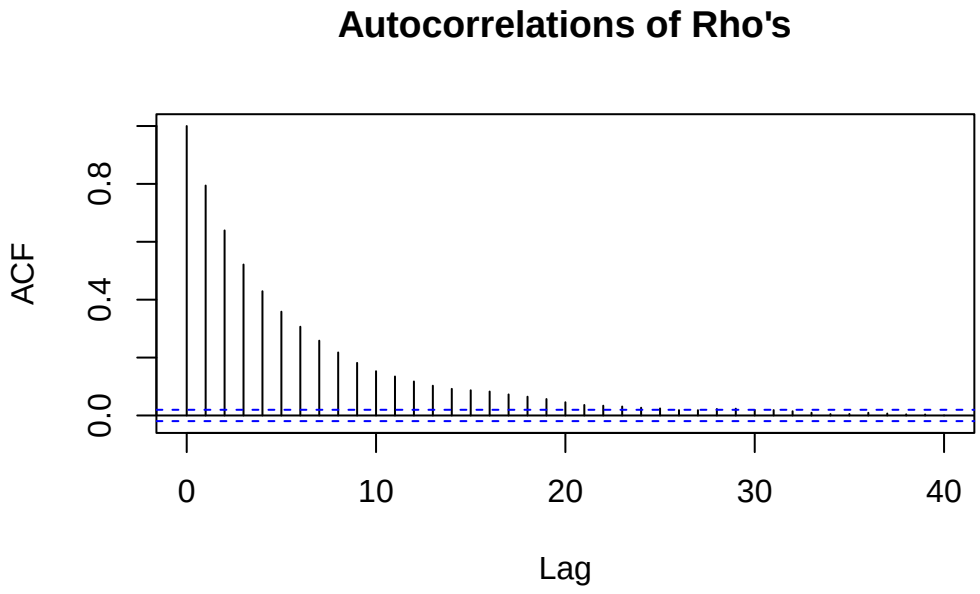
To evaluate the quality of this approximation, we begin with a trace plot. We see a nice thick band, indicating that our Metropolis-Hastings sampler is mixing well: draws of ρ never stay at the same level for too long, nor do they ever take too many consecutive steps in the same direction.

```
# Trace plot
plot(rho_values, type='l', main = "Trace Plot of Rho's")
```



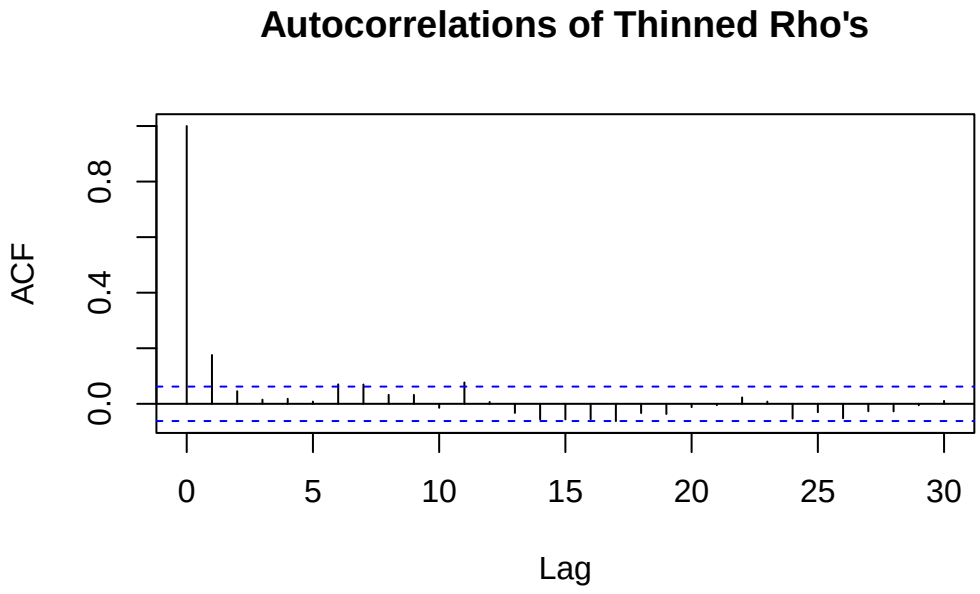
We further evaluate the quality of our approximation with an autocorrelation plot. Unfortunately, we see that autocorrelations fall outside of the desired bandwidth around 0 up to around the 20^{th} lag! This is a clear violation of the Markov chain structure we're looking for.

```
# Check autocorrelation
acf(rho_values, main = "Autocorrelations of Rho's") # not good!
```



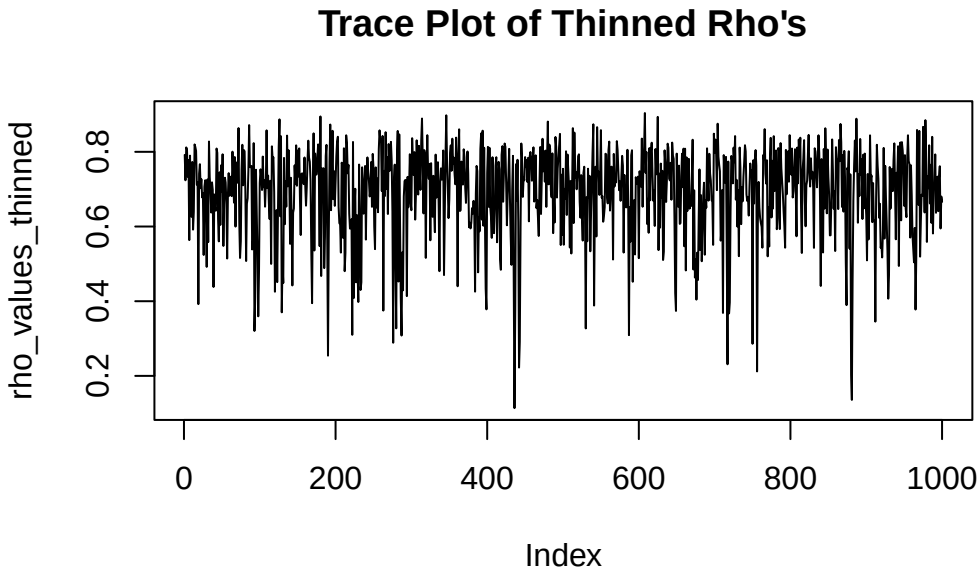
To manage this autocorrelation problem, we thin the data, taking every tenth draw of ρ. This leaves us with just 1,000 draws of ρ rather than the 10,000 we started with. The autocorrelations look much better now: there is a non-negligible lag-1 correlation (as desired), but the correlations across larger lags all fall within (or almost within, as in the case of the lag-11 autocorrelation) the desired bandwidth around 0.

```
# Thinning
rho_values_thinned = rho_values[10*(1:(nsim/10))]
acf(rho_values_thinned, main = "Autocorrelations of Thinned Rho's") #
 ↪  autocorrelation fixed!
```

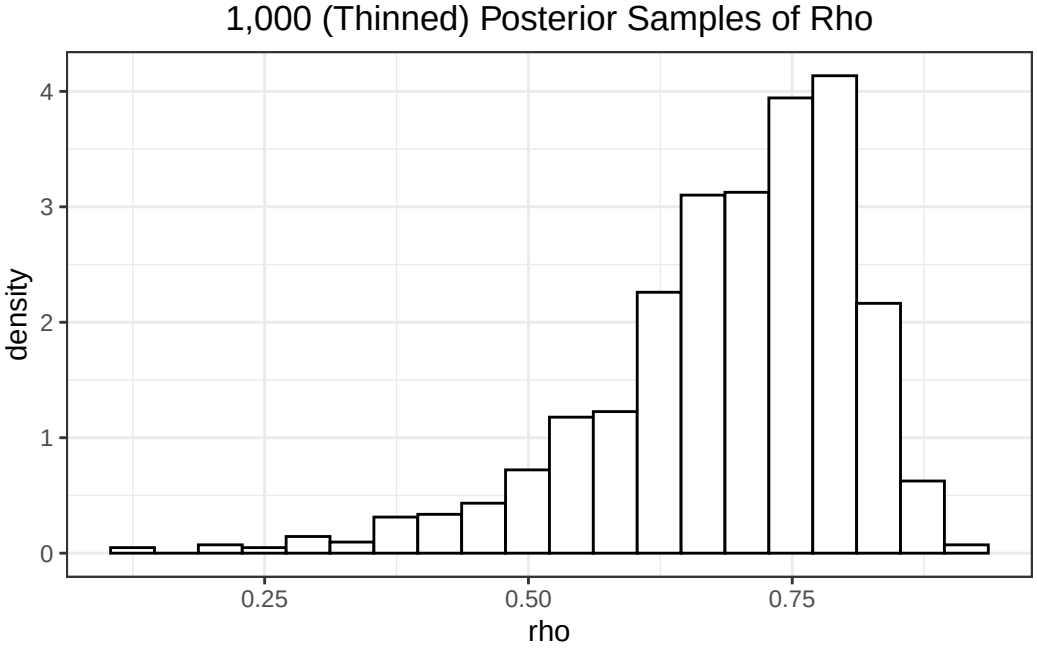


On the other hand, thinning causes the trace plot to suffer a little bit, though there is still a somewhat thick band, indicating that the algorithm is mixing well enough.

```
plot(rho_values_thinned, type='l', main = "Trace Plot of Thinned
 ↪  Rho's") # trace plot not too bad
```



We also plot the posterior density of ρ using the thinned set of values. Reassuringly, the histogram still resembles the histogram of all 10,000 simulated values of ρ, so thinning hasn't caused us to lose much information about the posterior distribution of ρ.



This (thinned) posterior distribution has a median value of $\rho_{med} \approx 0.717$ and a mean value of $\rho_{mean} \approx 0.690$. A quantile-based 95% credible interval for ρ is approximately

$(0.379, 0.857)$.

```
# Summarize the posterior distribution -- thinned data

# Central Tendency
median(rho_values_thinned)
```

```
[1] 0.717207
```

```
mean(rho_values_thinned)
```

```
[1] 0.690069
```

```
# quantile-based 95% credible interval
quantile(rho_values_thinned, probs=c(0.025, 0.975))
```

```
     2.5%     97.5%
0.3789015 0.8566189
```

2 Exercise 2

The Gibbs sampler can take a long time to converge if the target distribution is multi-modal. Suppose we are trying to sample from a posterior density that is the mixture of three normal densities: θ has density $0.45\mathcal{N}(3, \frac{1}{3}) + 0.10\mathcal{N}(0, \frac{1}{3}) + 0.45\mathcal{N}(-3, \frac{1}{3})$, where $\mathcal{N}(\mu, \sigma^2)$ represents the normal density with mean μ and variance σ^2. Consider using Gibbs sampling to sample from it.

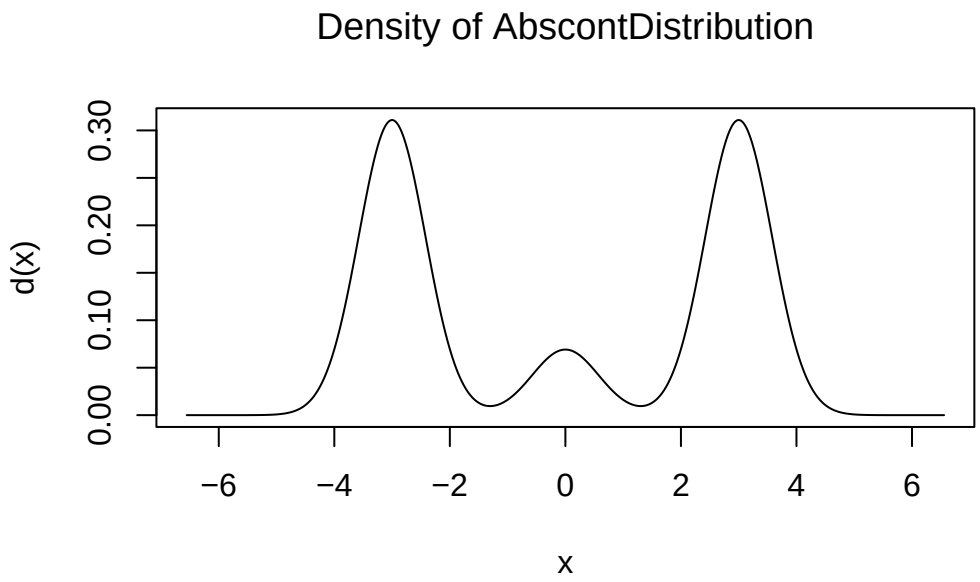
2.1 Part (a)

Plot the marginal density of θ. If the indicator $\delta \in \{1, 2, 3\}$, argue that the full conditional distribution for θ is $\theta|\delta \sim \mathcal{N}(\mu_\delta, \sigma^2_\delta)$. What are μ_δ and σ^2_δ here, for $\delta = 1, 2, 3$?

We plot the marginal density of θ, $0.45\mathcal{N}(3, \frac{1}{3}) + 0.10\mathcal{N}(0, \frac{1}{3}) + 0.45\mathcal{N}(-3, \frac{1}{3})$, below:

```
NormMix = UnivarMixingDistribution(Norm(mean=3, sd=sqrt(1/3)),
                                   Norm(mean=0, sd=sqrt(1/3)),
                                   Norm(mean=-3, sd=sqrt(1/3)),
                                   mixCoeff=c(0.45,0.1,0.45))
```

```r
plot(NormMix, to.draw.arg = "d")
```



By inspection, we guess that

$$\delta = \begin{cases} 1 & \text{w/p } 0.45 \\ 2 & \text{w/p } 0.10 \\ 3 & \text{w/p } 0.45 \end{cases},$$

so that the PMF of δ is

$$f(\delta) = \begin{cases} 0.45, & \delta = 1 \\ 0.10, & \delta = 2 \\ 0.45, & \delta = 3 \end{cases}.$$

Since $\theta \sim 0.45\mathcal{N}(3, \frac{1}{3}) + 0.10\mathcal{N}(0, \frac{1}{3}) + 0.45\mathcal{N}(-3, \frac{1}{3})$, the posterior density of θ is

$$f(\theta) = 0.45 \cdot \frac{1}{\sqrt{2\pi/3}} \exp\left\{-\frac{1}{2/3}(\theta-3)^2\right\} + 0.10 \cdot \frac{1}{\sqrt{2\pi/3}} \exp\left\{-\frac{1}{2/3}\theta^2\right\} + 0.45 \cdot \frac{1}{\sqrt{2\pi/3}} \exp\left\{-\frac{1}{2/3}(\theta+3)^2\right\}.$$

But by the properties of conditional probability, we also have

$$
\begin{aligned}
f(\theta) &= f(\delta) \times f(\theta|\delta) \\
&= \sum_{j=1}^{3} f(\delta = j) \times f(\theta|\delta = j) \\
&= f(\delta = 1) \times f(\theta|\delta = 1) + f(\delta = 2) \times f(\theta|\delta = 2) + f(\delta = 3) \times f(\theta|\delta = 3) \\
&= 0.45 f(\theta|\delta = 1) + 0.10 f(\theta|\delta = 2) + 0.45 f(\theta|\delta = 3).
\end{aligned}
$$

Comparing these two formulations for $f(\theta)$, we see that they share the same coefficients, and so the terms attached to corresponding coefficients must be the same. That is, it must be that

$$
\begin{aligned}
f(\theta|\delta = 1) &= \frac{1}{\sqrt{2\pi/3}} \exp\left\{ -\frac{1}{2/3}(\theta - 3)^2 \right\}, \\
f(\theta|\delta = 2) &= \frac{1}{\sqrt{2\pi/3}} \exp\left\{ -\frac{1}{2/3}\theta^2 \right\}, \text{ and} \\
f(\theta|\delta = 3) &= \frac{1}{\sqrt{2\pi/3}} \exp\left\{ -\frac{1}{2/3}(\theta + 3)^2 \right\}.
\end{aligned}
$$

That is, the full conditional distributions are

$$
\begin{aligned}
\theta|\delta = 1 &\sim \mathcal{N}\Big(3, \frac{1}{3}\Big), \\
\theta|\delta = 2 &\sim \mathcal{N}\Big(0, \frac{1}{3}\Big), \text{ and} \\
\theta|\delta = 3 &\sim \mathcal{N}\Big(-3, \frac{1}{3}\Big).
\end{aligned}
$$

More concisely, the full conditional distributions are

$$
\theta|\delta \sim \mathcal{N}(\mu_\delta, \sigma_\delta^2),
$$

as desired, where $\mu_1 = 3$, $\mu_2 = 0$, $\mu_3 = -3$, and $\sigma_1^2 = \sigma_2^2 = \sigma_3^2 = \frac{1}{3}$.

(Note that since $f(\delta = 1) = f(\delta = 3) = 0.45$, "switching" the conditional distributions associated with $\delta = 1$ and $\delta = 3$ would also be a valid solution.)

2.2 Part (b)

Use Bayes' Theorem to show that the full conditional for δ is

$$
\mathbb{P}(\delta = k|\theta) = \frac{\mathbb{P}(\delta = k) \times \mathcal{N}(\mu_k, \sigma_k^2)}{\sum_{j=1}^{3} \mathbb{P}(\delta = j) \times \mathcal{N}(\mu_j, \sigma_j^2)},
$$

for $k \in \{1, 2, 3\}$.

By Bayes' Theorem,

$$\begin{aligned}\mathbb{P}(\delta = k|\theta) &= \frac{f(\theta|\delta = k)\mathbb{P}(\delta = k)}{f(\theta)} \\ &= \frac{\mathbb{P}(\delta = k)f(\theta|\delta = k)}{\sum_{j=1}^{3} f(\theta|\delta = j)\mathbb{P}(\delta = j)} \\ &= \frac{\mathbb{P}(\delta = k) \times \mathcal{N}(\mu_k, \sigma_k^2)}{\sum_{j=1}^{3} \mathbb{P}(\delta = j) \times \mathcal{N}(\mu_j, \sigma_j^2)},\end{aligned}$$

as desired, where the second equality is by the Law of Total Probability, and the third equality is by our results in Part (a).

2.3 Part (c)

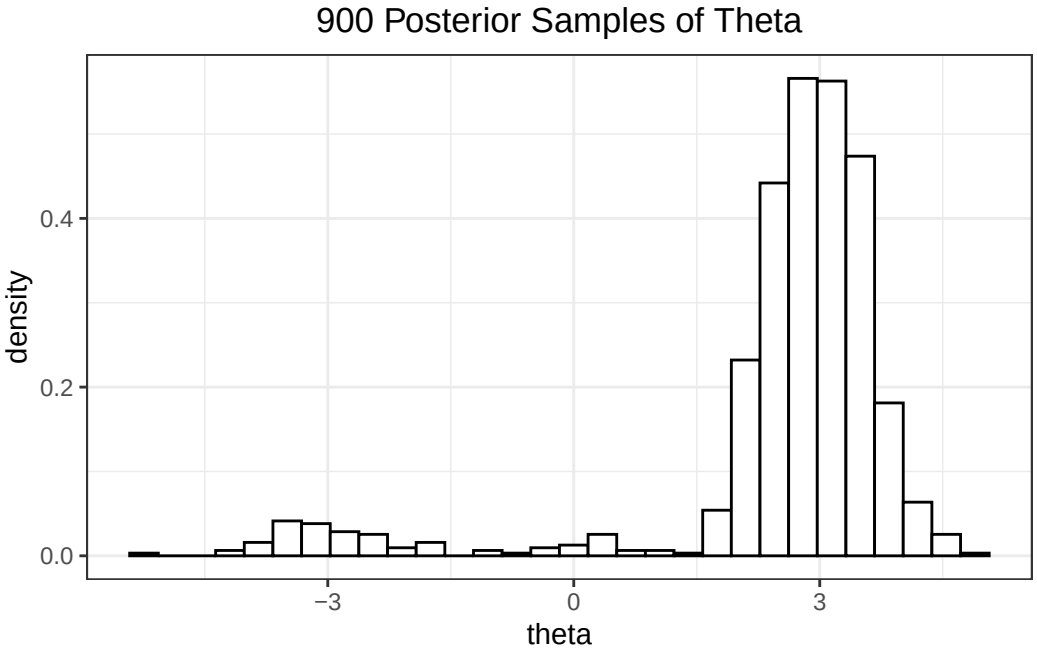
Write a Gibbs sampling algorithm to sample from the joint density of (θ, δ). Begin the chain with the initial values $\delta^0 = 2$ and $\theta^0 = 0$, and generate 1,000 values of θ. Give a plot of a relative frequency histogram of the θ values (using a command like `hist(theta.values, freq=F)` in R) and comment on how it compares to the true marginal density of θ plotted in Part (a).

First, we write a function that will handle our simulations in this question as well as in Part (d).

```
q2sampler = function(nsim, nburn){
  # Create empty arrays that will store parameter values
  thetas = rep(0,nsim)
  deltas = rep(0,nsim)

  # Initialize based on delta0 = 2, theta0 = 0
  thetas[1] = 0
  deltas[1] = 2

  # Gibbs algo
  for(j in 2:nsim){
    # First, draw theta from a Normal dist based on previous
     ↪  iteration's delta
    if(deltas[j-1] == 1){
      thetas[j] = rnorm(1, 3, sqrt(1/3))
    }
    else if(deltas[j-1] == 2){
      thetas[j] = rnorm(1, 0, sqrt(1/3))
```

```
    }
    else if(deltas[j-1] == 3){
      thetas[j] = rnorm(1, -3, sqrt(1/3))
    }

    # Second, draw delta based on this iteration's value of theta
    ### Denominator of Bayes Thm in Part (b)
    denom = 0.45*dnorm(thetas[j], mean = 3, sd = sqrt(1/3)) +
            0.10*dnorm(thetas[j], mean = 0, sd = sqrt(1/3)) +
            0.45*dnorm(thetas[j], mean = -3, sd = sqrt(1/3))
    ### Conditional probs of delta=k given this iteration's theta
    prob1 = 0.45*dnorm(thetas[j], mean = 3, sd = sqrt(1/3)) / denom
    prob2 = 0.10*dnorm(thetas[j], mean = 0, sd = sqrt(1/3)) / denom
    prob3 = 0.45*dnorm(thetas[j], mean = -3, sd = sqrt(1/3)) / denom
    ### Uniformly generate a random # from [0,1]
    rand = runif(1, 0, 1)
    ### Pick delta based on uniform draw and conditional probs
    if(rand <= prob1){
      deltas[j] = 1
    }
    else if(rand > prob1 & rand <= prob1 + prob2){
      deltas[j] = 2
    }
    else if(rand > prob1 + prob2 & rand <= prob1 + prob2 + prob3){
      deltas[j] = 3
    }
  }

  # Just return the values of theta & delta from after the burn-in
   ↪ period
  postburn_thetas = thetas[nburn+1:nsim]
  postburn_deltas = deltas[nburn+1:nsim]
  return(matrix(c(postburn_thetas, postburn_deltas), ncol = 2))
}
```

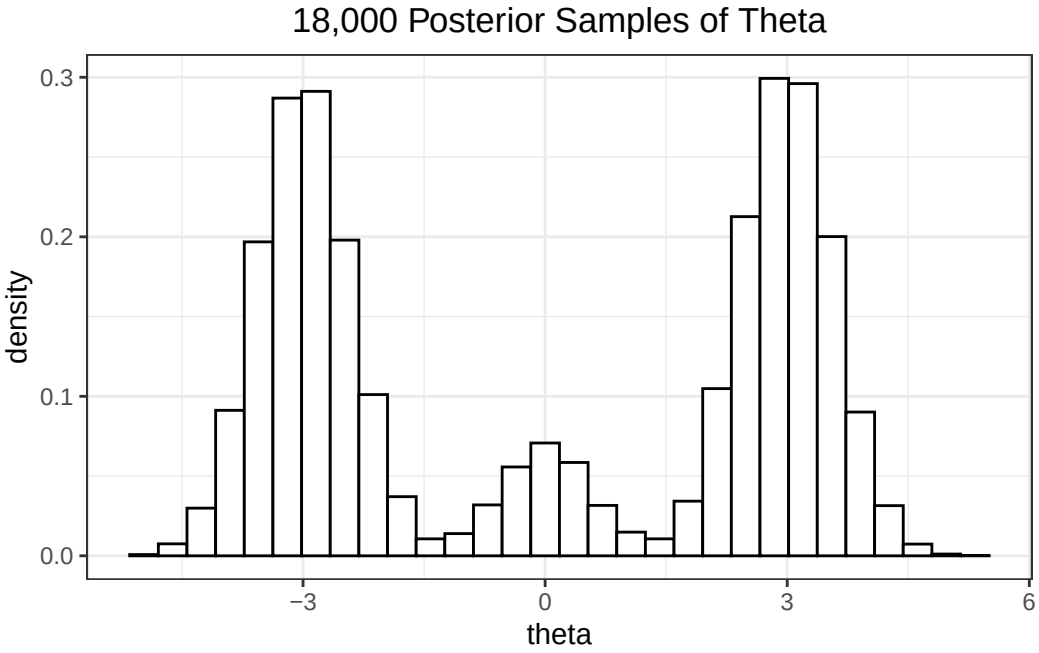
We generate 1,000 values of θ, throw out the first 10% (to account for the burn-in period), and plot the relative frequencies of the remaining 900 values in the histogram below. We can see the very beginnings of a multi-modal distribution forming, with small local peaks around $\mu_3 = -3$ and $\mu_2 = 0$. However, the vast majority of the density of θ is concentrated around $\mu_1 = 3$. It seems that with only 1,000 iterations, the Gibbs sampling algorithm has not had enough time to fully explore the range of values θ can take, and in particular has barely sampled from the left and center peaks of θ's posterior distribution.



2.4 Part (d)

Repeat Part (c), but generate 20,000 values of θ. Again give a plot of a relative frequency histogram of the θ values and comment on how it compares to the true marginal density of θ plotted in Part (a).

We generate 20,000 samples of θ, throw out the first 10% (to account for the burn-in period), and plot the relative frequencies of the remaining 18,000 values in the histogram below. Now, the histogram looks just like the true posterior density of θ we plotted in Part (a): the histogram has local modes around $\mu_3 = -3$, $\mu_2 = 0$, and $\mu_1 = 3$. Moreover, the densities of the left and right modes are both around 0.3 (just as they were in Part (a)), and the density of the center mode is around 0.075 (just as it was in Part (a)). That is, with 20,000 iterations, the Gibbs sampling algorithm has not only qualitatively captured the shape of the posterior distribution of θ, but has even quantitatively captured the right densities of different values of θ (approximately).



3 Exercise 3

Suppose that $D = 5$ doses of a medication are being tested for toxicity. When a patient is put on a dose, they might experience a toxicity event. Suppose n_i patients have been treated using dose i and x_i of them experience toxicity. The data are

Dose d_i	1	2	3	4	5
n_i	3	3	3	6	3
x_i	0	0	0	3	2

Assume $x_i|n_i \sim \text{Bin}(n_i, p_i)$ and apply the logistic regression

$$p_i = \frac{\exp(\beta_0 + \beta_1 d_i)}{1 + \exp(\beta_0 + \beta_1 d_i)}.$$

Suppose $\beta_0 = -3$ and let β_1 follow a prior $\mathcal{N}(0, 2)$ distribution. Use Metropolis-Hastings to sample the posterior $p(\beta_1|\{n_i, x_i\})$, where $\{n_i, x_i\}$ represent the data on all five doses. Plot the histogram of β_1 posterior samples, and obtain the posterior mean and variance of β_1. What is the value of

$$\hat{p}_4 = \frac{\exp(\beta_0 + \hat{\beta}_1 d_4)}{1 + \exp(\beta_0 + \hat{\beta}_1 d_4)},$$

where $\hat{\beta}_1$ is the posterior mean?

Since $\beta_1 \sim \mathcal{N}(0, 2)$, the prior density of β_1 is

$$\begin{aligned} f(\beta_1) &= \frac{1}{\sqrt{2\pi(2)}} \exp\left\{ -\frac{1}{2(2)}(\beta_1 - 0)^2 \right\} \\ &= \frac{1}{\sqrt{4\pi}} \exp\{-\beta_1^2/4\}. \end{aligned}$$

Moreover, since $x_i|n_i, d_i \sim \text{Bin}(n_i, p_i)$ with

$$p_i = \frac{\exp(\beta_0 + \beta_1 d_i)}{1 + \exp(\beta_0 + \beta_1 d_i)} := \text{expit}(\beta_0 + \beta_1 d_i) = \text{expit}(-3 + \beta_1 d_i),$$

the likelihood function is

$$\begin{aligned} \mathcal{L}(\beta_1|\text{x}, \text{n}) &= \prod_{i=1}^{5} f(x_i|n_i, d_i) \\ &= \prod_{i=1}^{5} \text{nCr}(n_i, x_i) \cdot p_i^{x_i}(1 - p_i)^{n_i - x_i} \\ &= \prod_{i=1}^{5} \text{nCr}(n_i, x_i) \cdot \text{expit}(-3 + \beta_1 d_i)^{x_i}[1 - \text{expit}(-3 + \beta_1 d_i)]^{n_i - x_i} \\ &= \text{nCr}(3, 0) \cdot \text{expit}(-3 + \beta_1)^0[1 - \text{expit}(-3 + \beta_1)]^{3-0} \\ &\quad * \text{nCr}(3, 0) \cdot \text{expit}(-3 + 2\beta_1)^0[1 - \text{expit}(-3 + 2\beta_1)]^{3-0} \\ &\quad * \text{nCr}(3, 0) \cdot \text{expit}(-3 + 3\beta_1)^0[1 - \text{expit}(-3 + 3\beta_1)]^{3-0} \\ &\quad * \text{nCr}(6, 3) \cdot \text{expit}(-3 + 4\beta_1)^3[1 - \text{expit}(-3 + 4\beta_1)]^{6-3} \\ &\quad * \text{nCr}(3, 2) \cdot \text{expit}(-3 + 5\beta_1)^2[1 - \text{expit}(-3 + 5\beta_1)]^{3-2} \\ &= [1 - \text{expit}(-3 + \beta_1)]^3 * [1 - \text{expit}(-3 + 2\beta_1)]^3 * [1 - \text{expit}(-3 + 3\beta_1)]^3 \\ &\quad * 20 \cdot \text{expit}(-3 + 4\beta_1)^3[1 - \text{expit}(-3 + 4\beta_1)]^3 \\ &\quad * 3 \cdot \text{expit}(-3 + 5\beta_1)^2[1 - \text{expit}(-3 + 5\beta_1)]. \end{aligned}$$

We sample from the posterior distribution of $\beta_1|\mathbf{x}, \mathbf{n}$ using a Metropolis-Hastings algorithm that incorporates the above prior and likelihood, has 10,000 iterations, sets an initial value of $\beta_1^{(1)} = 0$ and uses the random walk proposal density $\beta_1^{(proposed)} = \beta_1^{(t-1)} + \varepsilon$, $\varepsilon \sim \mathcal{N}(0, 0.2)$. As we showed in class, this is a symmetric proposal density, which we exploit in our algorithm.

```
nsim = 10000 # iterations of algo
beta11 = 0 # initial value of beta1
accept = 0 # counter for acceptance rate
```

```r
beta1_values = matrix(0, nrow = nsim) # store beta1 values
beta1_values[1] = beta11

set.seed(41) # favorite number

# expit function
expit = function(x){
  return(exp(x)/(1+exp(x)))
}

# Likelihood function
likelihood = function(beta1){
  (1-expit(-2+beta1))^3 *
    (1-expit(beta0+2*beta1))^3 *
    (1-expit(beta0+3*beta1))^3 *
    20*expit(beta0+4*beta1)^3*(1-expit(beta0+4*beta1))^3 *
    3*expit(beta0+5*beta1)^2*(1-expit(beta0+5*beta1))
}

# MH algorithm
for(i in 2:nsim){
  # Step 1: Draw beta1_prop from proposal density: random walk N(0,1/5)
  ↪ (symm.)
  beta1_prop = beta1_values[i-1] + rnorm(1, 0, 1/5)

  # Step 2: Acceptance/Odds ratio
  ### Logs of priors
  prior_beta1_prop = dnorm(beta1_prop, mean = 0, sd = sqrt(2), log = T)
  prior_beta1_old = dnorm(beta1_values[i-1], mean = 0, sd = sqrt(2),
 ↪ log = T)

  ### Logs of likelihoods
  lik_beta1_prop = log(likelihood(beta1_prop))
  lik_beta1_old = log(likelihood(beta1_values[i-1]))

  ### Acceptance ratio
  log_accept_ratio = (prior_beta1_prop + lik_beta1_prop) -
                     (prior_beta1_old + lik_beta1_old)
  accept_ratio = exp(log_accept_ratio)

  # Steps 3/4: Accept/Reject
  if(accept_ratio >= 1){
    # auto-accept if r >= 1
```

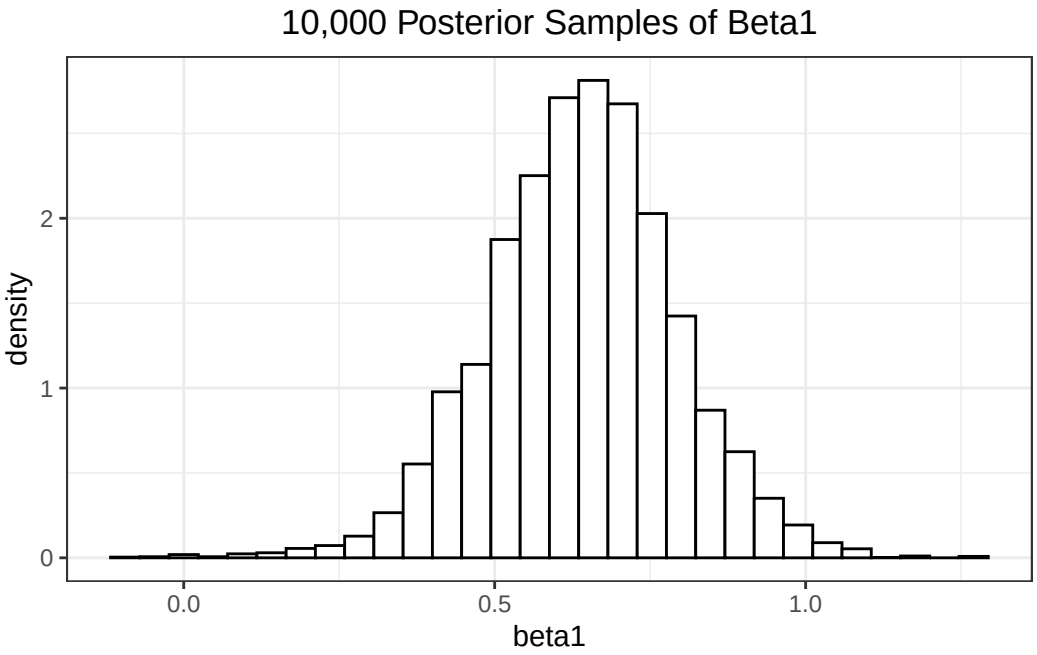
```
    beta1_values[i] = beta1_prop
    accept = accept + 1
  }
  else if(accept_ratio < 1){
    # Draw random number from [0,1]
    draw = runif(1, 0, 1)
    if(draw <= accept_ratio){
      # Accept
      beta1_values[i] = beta1_prop
      accept = accept + 1
    }
    else{
      # Reject
      beta1_values[i] = beta1_values[i-1]
    }
  }
}
```

Our algorithm's acceptance rate was 62.15%, which falls squarely in the desirable $[20\%, 80\%]$ range for convergence.

```
accept/nsim # ~62%
```

```
[1] 0.6215
```

We plot the posterior density of β_1 based on our Metropolis-Hastings algorithm below:



We estimate that $\mathbb{E}[\beta_1|\mathbf{n},\mathbf{x}] \approx 0.640$, and $\text{Var}(\beta_1|\mathbf{n},\mathbf{x}) \approx 0.023$.

```
mean(beta1_values)
```

```
[1] 0.6401395
```

```
var(beta1_values) * (length(beta1_values)-1)/length(beta1_values) #
 ↪  pop'n variance
```

```
           [,1]
[1,] 0.02308435
```

We also estimate that

$$\begin{aligned}\hat{p}_4 &= \text{expit}(\beta_0 + \hat{\beta}_1 d_4) \\ &\approx \text{expit}(-3 + 0.640 \times 4) \\ &\approx 0.392.\end{aligned}$$

```
expit(-3+mean(beta1_values)*4)
```

```
[1] 0.3918739
```