Lab 8: Gibbs Sampling

Isaac Fan

3/19/2021

```
library(readr)
mystery <- read_csv("C:/Users/isaac/Downloads/mystery.csv")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   x = col_double()
## )
```

Preliminaries

Please turn in a pdf copy of the RMD document by Friday, March 26 at 11:59 PM. Only exercises 2.1, 2.2, and 2.3 will be graded (for completion), but you should try all of the exercises (especially example 3). Any derivations can be submitted by taking a picture or scan of written work. You will have most of the lab to work on the exercises and to ask questions.

Introduction

Suppose we have a random vector $\boldsymbol{X} = (X_1, \ldots, X_p)^T$ and we want to generate samples from its joint distribution. The *Gibbs Sampler* is a Markov process which utilizes sampling from specific conditional distributions, termed *full conditionals*, to sample from a target joint distribution. Under certain regularity conditions, which are satisfied by all of the examples below (and most likely all the examples you'll see in class), the Gibbs sampler is a stationary Markov process. In other words, iteratively sampling from the full conditional distributions of $\boldsymbol{X}$ is equivalent to generating a sample from the joint distribution of $\boldsymbol{X}$.

Contrary to some of the Monte Carlo methods you have seen so far, samples from the Gibbs sampler are correlated. This means that we need to use diagnostics, such as ACF plots, effective sample size, and traceplots, to gauge the quality of your sample. Another consideration is that you need to sample from the full conditional distributions themselves. Thinking about these full conditional distributions can be complicated as they require fully understanding the *structure* of a model. Fortunately, once you understand the model's structure, the full conditional derivations tend to follow from some of the derivations you have done on past homework assignments and in class.

In this lab, we'll work through three separate examples. Note that only example 3 involves a Bayesian model. This is by design, as Gibbs sampling was invented to sample from arbitrary joint distributions.

Example 1: Bivariate Normal

Consider the bivariate random vector (x, y) with distribution

$$\begin{aligned} x &\sim \mathcal{N}(0,1); \\ y \mid x &\sim \mathcal{N}(\rho x, 1-\rho^2) \end{aligned}$$

for some $\rho \in (-1, 1)$.

Exercise 1.1

1. Derive the full conditional distribution for $y|x$.
2. Derive the full conditional distribution for $x|y$.

Exercise 1.2

What is the marginal distribution for y? *Hint: you don't have to do an integral.*

Example 2: Trivariate Poisson-Gamma

Let (x, y, z) be a trivariate random vector with distribution

$$\begin{aligned} x &\sim \text{Gamma}(a, b); \\ y \mid x &\sim \text{Exp}(x); \\ z \mid x, y &\sim \text{Poisson}(y) \end{aligned}$$

for some $a, b > 0$.

Exercise 2.1

1. Derive the full conditional distribution for $z \mid x, y$.
2. Derive the full conditional distribution for $y \mid x, z$.
3. Derive the full conditional distribution for $x \mid y, z$.

Exercise 2.2

Using the skeleton code in the RMD document, write a Gibbs sampler for sampling from $p(x, y, z)$. Run the Gibbs sampler for $R = 10000$ iterations with $a = 5$ and $b = 2$.

```r
trivGibbs <- function(R, a, b) {
  # R = number of iterations
  # a,b = hyperparameters for x

  # storage and initialization
  x = rep(1, R)
  y = rep(1, R)
  z = rep(1, R)
  # sample from the full conditionals
  for (j in 2:R) {
    # sample from z | x, y
    z[j] <- rpois(1, y[j-1])
    # sample from y | x, z
    y[j] <- rgamma(1, z[j] + 1, x[j-1] + 1)
    # sample from x | y, z
    x[j] <- rgamma(1, a + 1, y[j] + b)
  }

 return(list(x = x, y = y, z = z))

}

g <- trivGibbs(10000, 5, 2)
```

Report posterior means and credible intervals, and make histograms for x, y, and z.

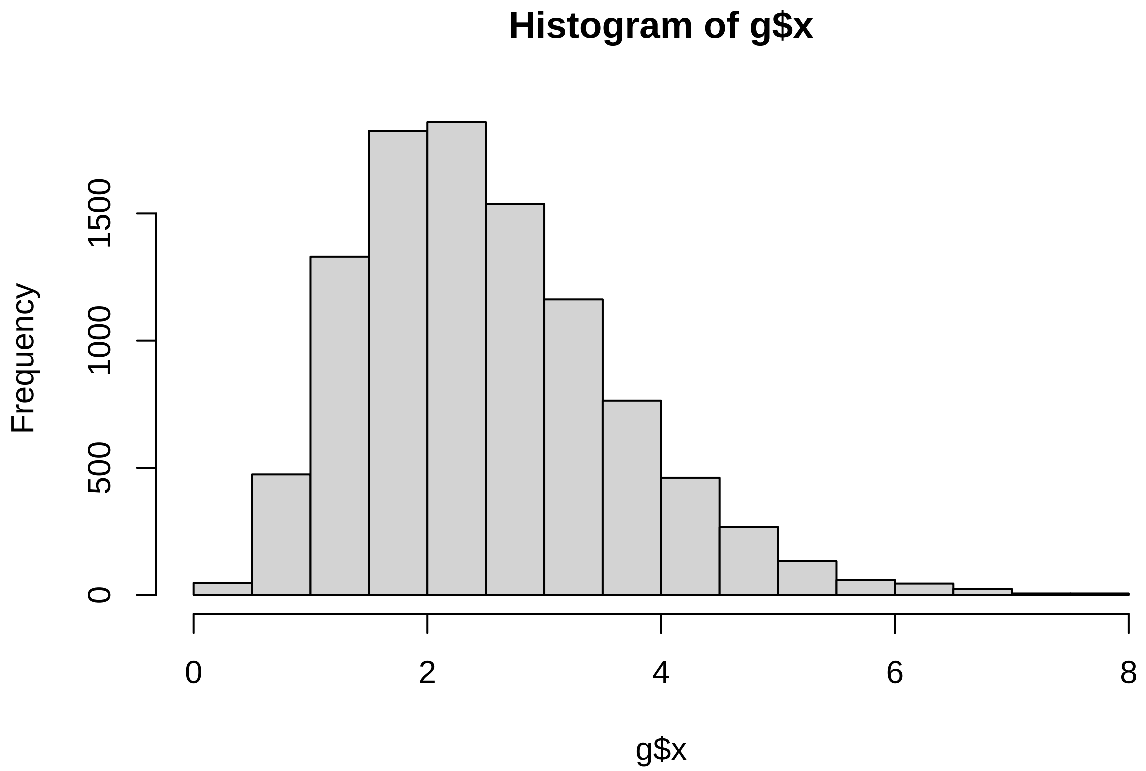
```r
#Mean, Credible Intervals, and Histogram for X
mean(g$x)
```

```
## [1] 2.507649
```

```r
c(mean(g$x) - 1.96 * sd(g$x), mean(g$x) + 1.96 * sd(g$x))
```

```
## [1] 0.3243735 4.6909239
```

```r
hist(g$x)
```



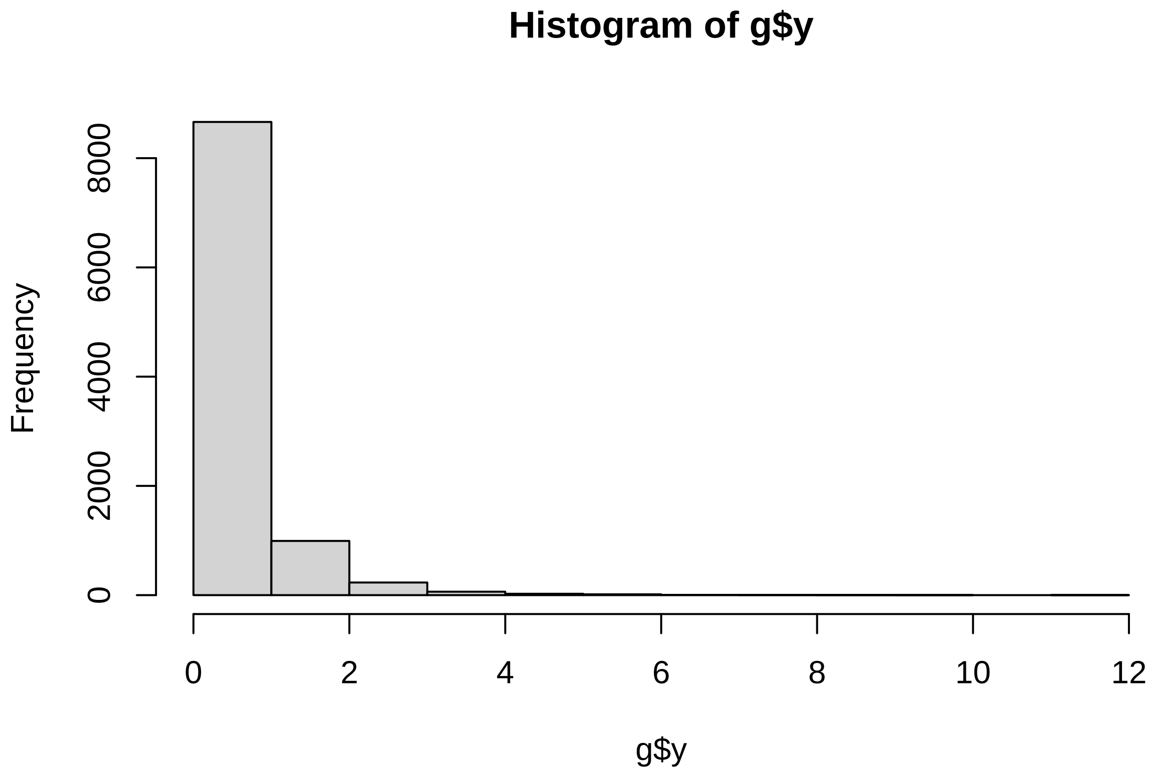
```
#Mean, Credible Intervals, and Histogram for Y
mean(g$y)
```

```
## [1] 0.5057938
```

```
c(mean(g$y) - 1.96 * sd(g$y), mean(g$y) + 1.96 * sd(g$y))
```

```
## [1] -0.8085149  1.8201024
```

```
hist(g$y)
```



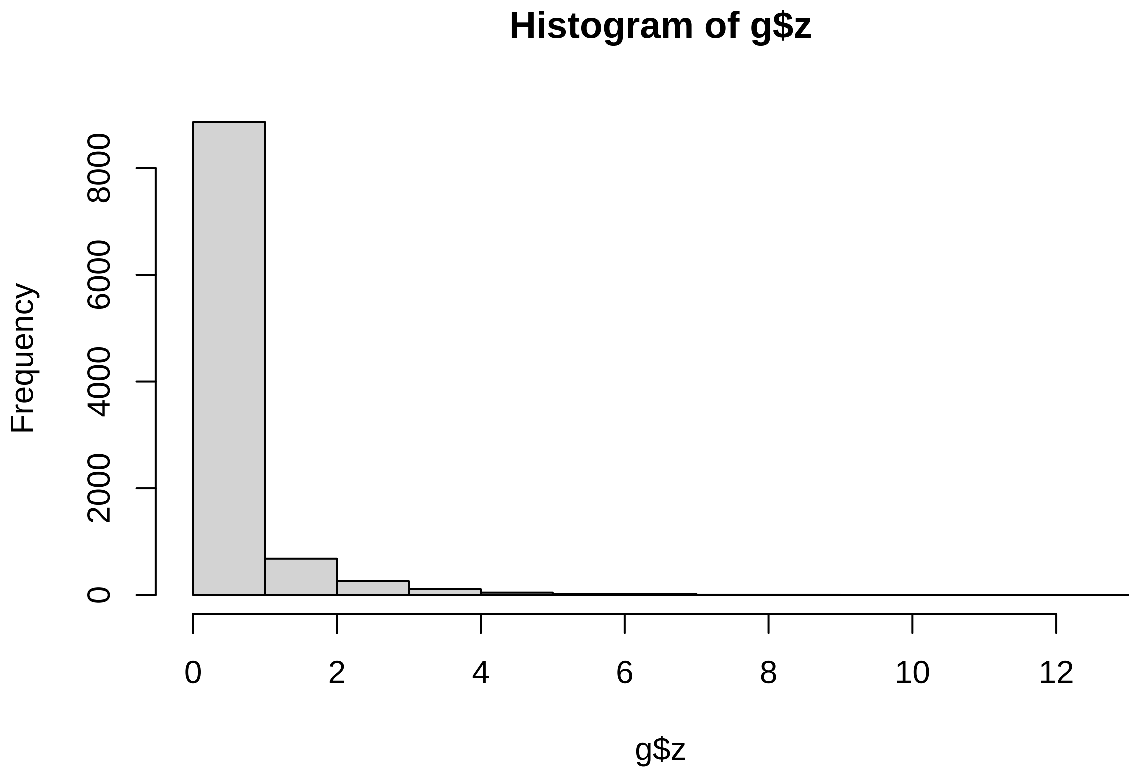
```
#Mean, Credible Intervals, and Histogram for Z
mean(g$z)
```

```
## [1] 0.5165
```

```
c(mean(g$z) - 1.96 * sd(g$z), mean(g$z) + 1.96 * sd(g$z))
```

```
## [1] -1.441762  2.474762
```

```
hist(g$z)
```



Exercise 2.3

For x, y, and z,

1. Calculate the effective sample size.

```
# ESS for X
effectiveSize(g$x)
```

```
##     var1
## 6445.625
```

```
# ESS for Y
effectiveSize(g$y)
```

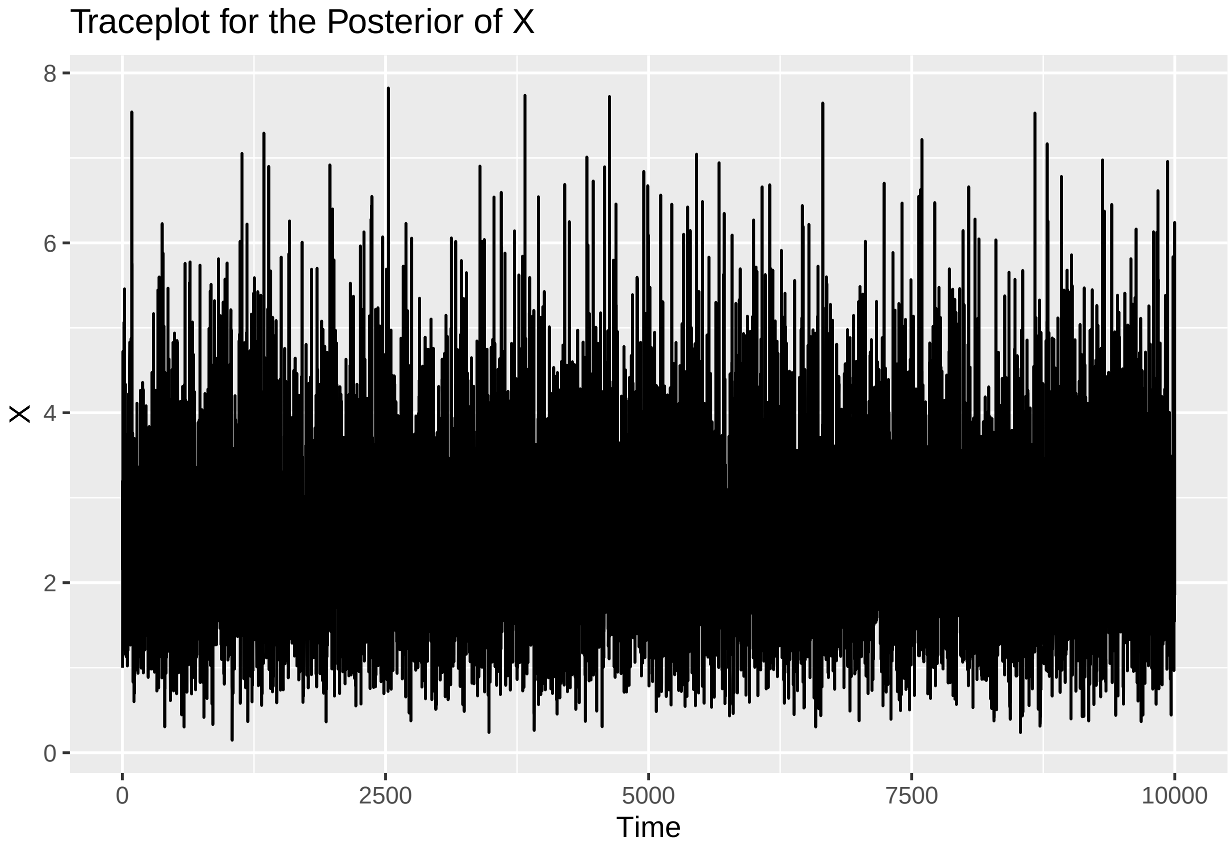
```
##     var1
## 2362.502
```

```
# ESS for Z
effectiveSize(g$z)
```

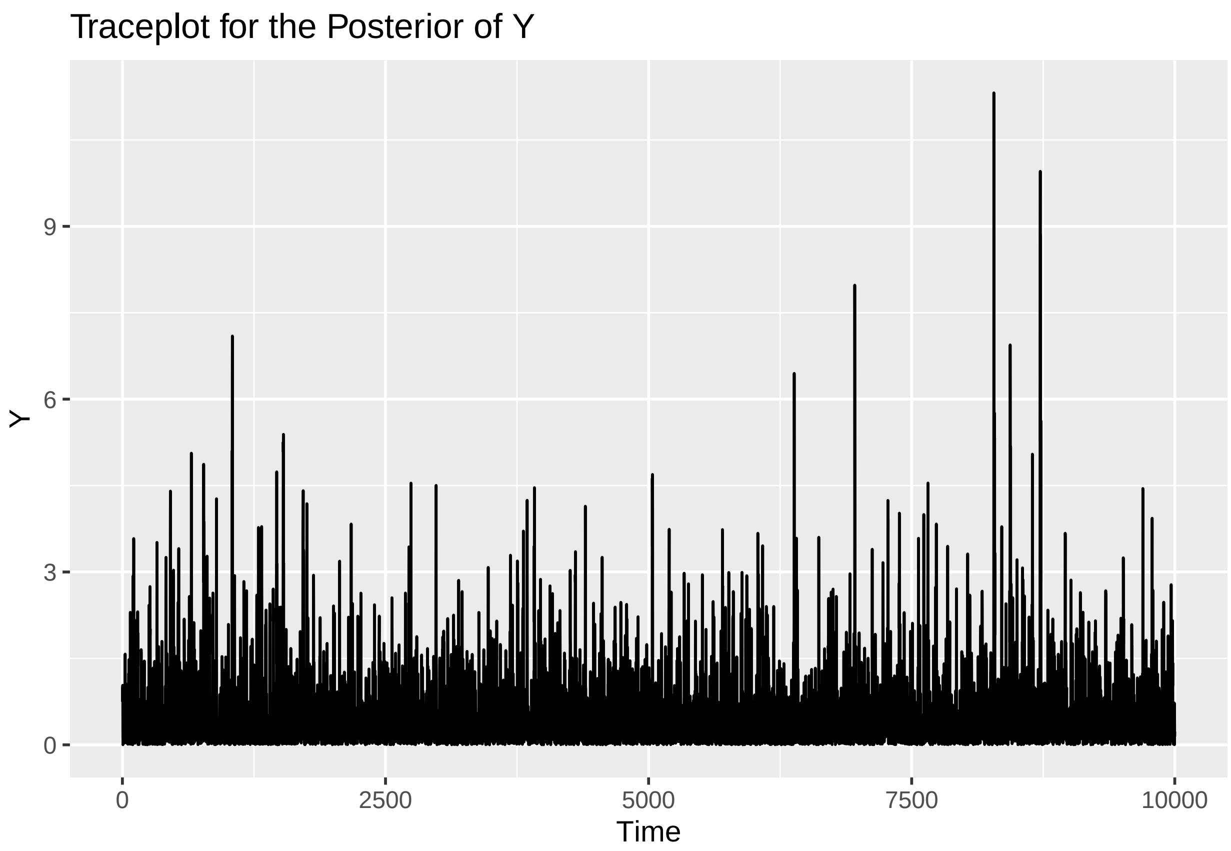
```
##     var1
## 2782.373
```

2. Create traceplots.

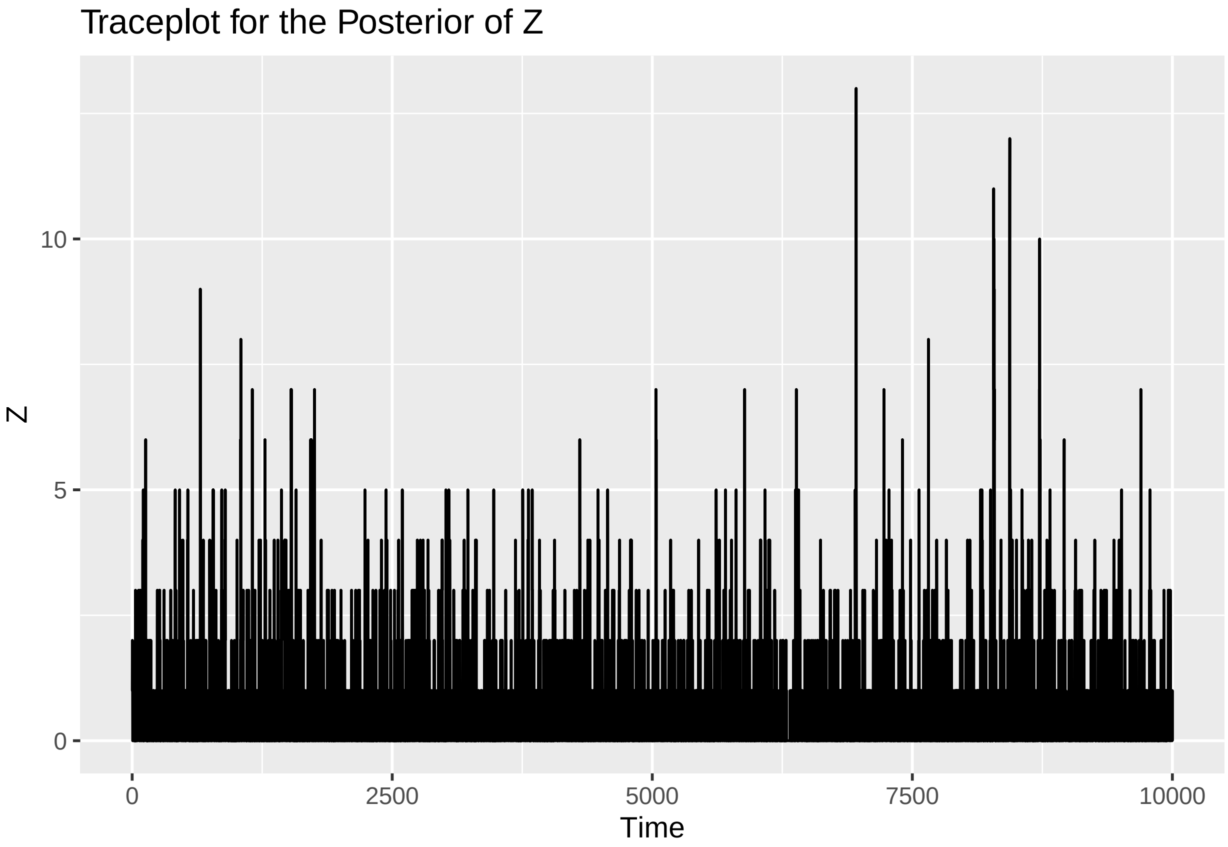
```
# traceplot for X
ggplot(data = data.frame(g), aes(x = 1:length(x), y = x)) + geom_line() + labs(x = "Time", y = "X", titl
```



```
# traceplot for Y
ggplot(data = data.frame(g), aes(x = 1:length(y), y = y)) + geom_line() + labs(x = "Time", y = "Y", titl
```

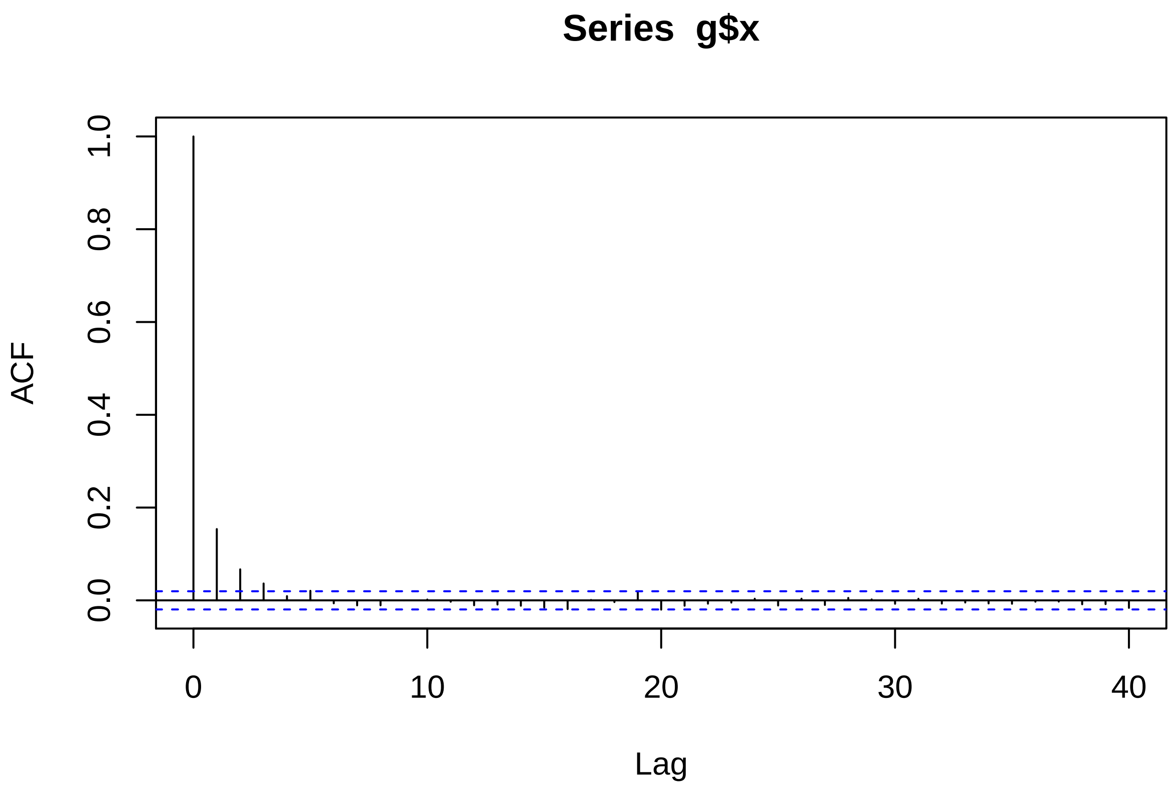


```r
# traceplot for Z
ggplot(data = data.frame(g), aes(x = 1:length(z), y = z)) + geom_line() + labs(x = "Time", y = "Z", titl
```

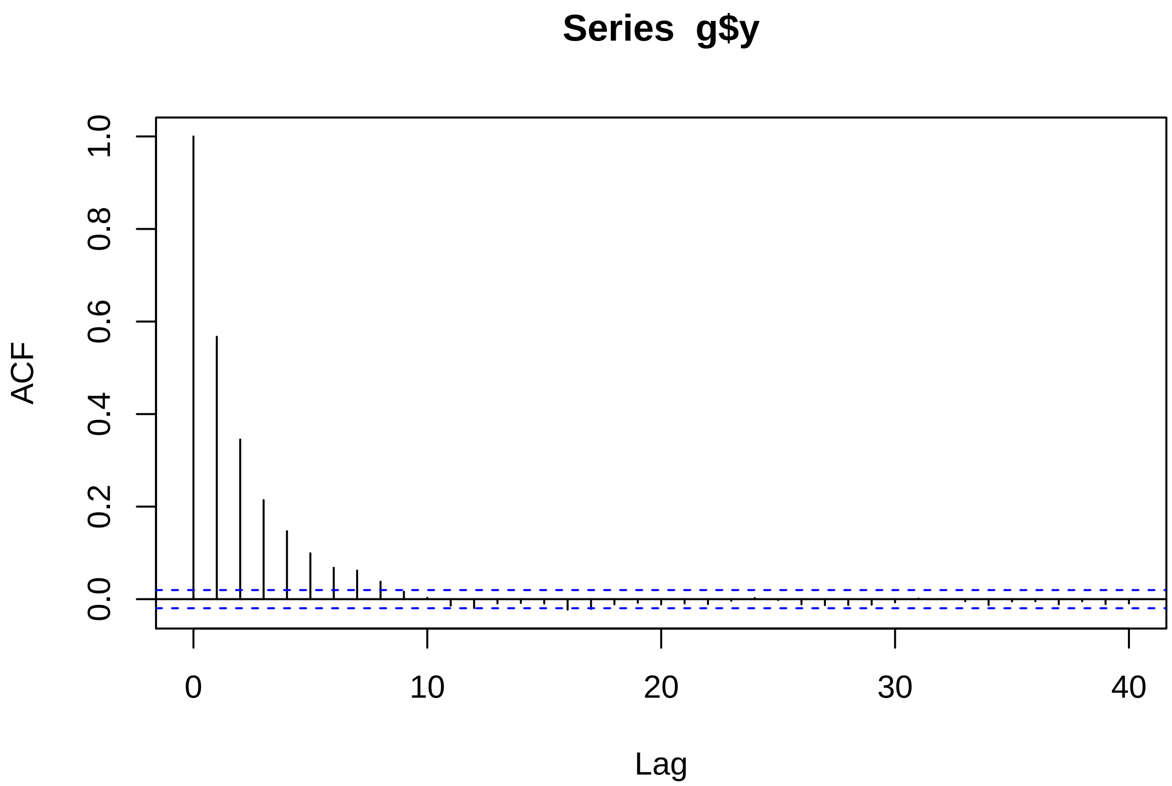


3. Create ACF plots.

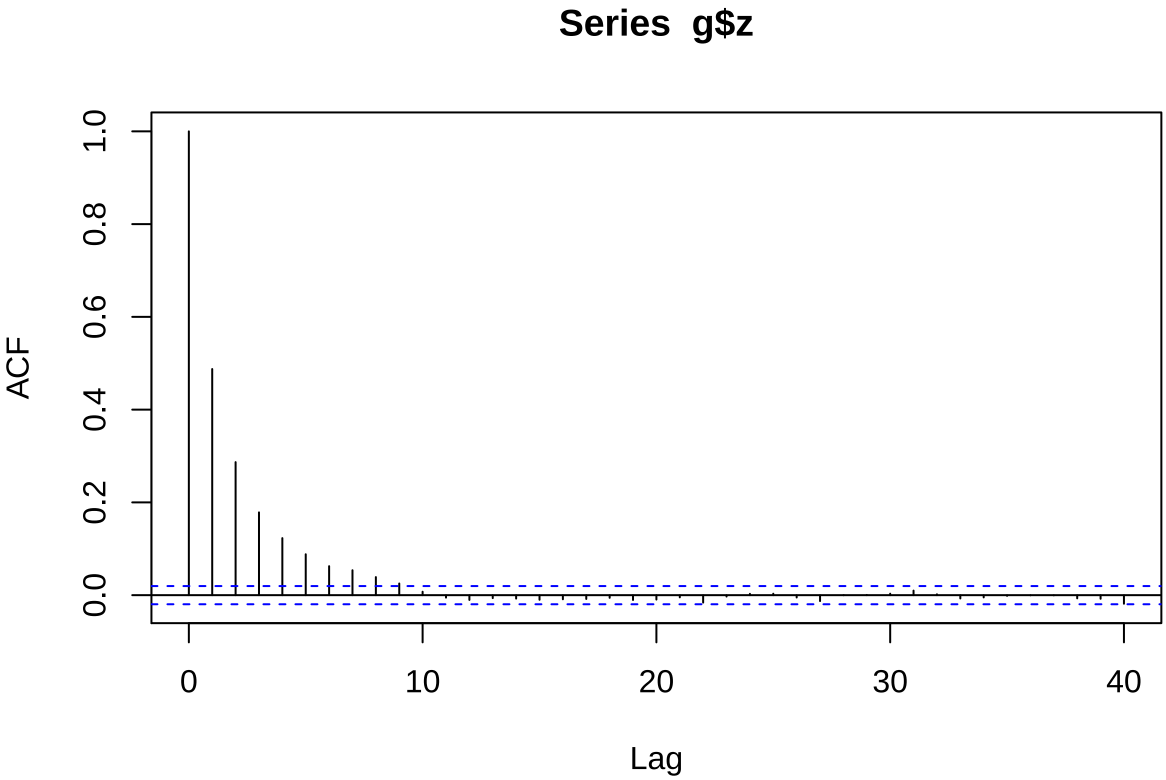
```
# ACF plot of X
acf(g$x)
```



```
# ACF plot of Y
acf(g$y)
```



```
# ACF plot of Z
acf(g$z)
```



\end{enumerate}

Example 3: Normal Mixture Model Posterior Distribution

Let

$$y_1, \dots, y_n \mid \pi, \mu_1, \mu_2 \overset{iid}{\sim} \text{NM}(\pi, \mu_1, \mu_2, 1), \tag{1}$$

a **normal mixture model** with density

$$f(y) = \pi \left\{ (2\pi)^{-1/2} \, e^{-\frac{1}{2}(y-\mu_1)^2} \right\} + (1-\pi) \left\{ (2\pi)^{-1/2} \, e^{-\frac{1}{2}(y-\mu_2)^2} \right\}$$

for $y \in \mathbb{R}$. We assign the following priors for π, μ_1, μ_2:

$$\pi \sim \text{Beta}(a, b); \tag{2}$$

$$\mu_j \sim \mathcal{N}(\theta_j, \tau_j^2); \tag{3}$$

$$\tag{4}$$

with independence structure

$$p(\pi, \mu_1, \mu_2) = p(\pi)p(\mu_1)p(\mu_2).$$

The joint probability density function of $(y_1, \dots, y_n)^T \mid \pi, \mu_1, \mu_2$ is very complicated. An equivalent model to (1) is formulated in terms of pairs (y_i, z_i) such that

$$z_1, \dots, z_n \mid \pi, \mu_1, \mu_2 \overset{iid}{\sim} \text{Bernoulli}(\pi) \tag{5}$$

$$y_i \mid z_i, \pi, \mu_1, \mu_2 \overset{ind}{\sim} \mathcal{N}(\mu_{z_i}, 1) \text{ for } i = 1, \dots, n. \tag{6}$$

In other words, when $z_i = 1$, y_i is sampled from the first normal distribution in the mixture. When $z_i = 0$, y_i is sampled from the second normal distribution. As it turns out, the introduction of the latent z_i variables leads to a straightforward Gibbs sampler for the distribution of $\pi, \mu_1, \mu_2 \mid \boldsymbol{y}$.

As you will see, these latent variables greatly reduce the complexity in our model when we condition on them. For example, $p(\mu_1 \mid \boldsymbol{y}, \pi, \mu_2)$ would be very difficult to derive due to the complicated likelihood function. However, $p(\mu_1 \mid \boldsymbol{y}, \boldsymbol{z}, \pi, \mu_2)$ almost trivially follows from some of the derivations you've seen in class. So, our strategy to sample from $p(\mu_1, \mu_2, \pi \mid y)$ is to sample from $p(\mu_1, \mu_2, \pi, \boldsymbol{z} \mid y)$ and then simply discard the $\boldsymbol{z}$ samples.

While this is very convenient, there is a catch: $\boldsymbol{z}$ is never observed. In other words, the vector $\boldsymbol{z}$ is treated as another unknown parameter in this model, where $\boldsymbol{z} \in \{0, 1\}^n$. Therefore, we also have to derive a full conditional for $\boldsymbol{z}$. With this in mind, we can now derive the Gibbs sampler.

Exercise 3.1

Derive the full conditional distributions for

1. $p(z_i \mid \boldsymbol{y}, z_{-i}, \pi, \mu_1, \mu_2)$. *Hint: there is lots of conditional independence in this model. Also, remember that $z_i \in \{0, 1\}$.*
2. $p(\mu_1 \mid \boldsymbol{y}, \boldsymbol{z}, \pi, \mu_2)$.
3. $p(\mu_2 \mid \boldsymbol{y}, \boldsymbol{z}, \pi, \mu_1)$.
4. $p(\pi \mid \boldsymbol{y}, \boldsymbol{z}, \mu_1, \mu_2)$.

Exercise 3.2

Write a Gibbs sampler to sample from $p(\mu_1, \mu_2, \pi, \boldsymbol{z} \mid \boldsymbol{y})$ using the skeleton code provided in the RMD document.

```r
mixGibbs <- function(R, a, b, theta_1, tau_1, theta_2, tau_2, y) {
  # R = number of iterations
  # a,b = prior parameters for pi
  # theta_1, tau_1 = prior parameters for mu_1
  # theta_2, tau_2 = prior parameters for mu_2
  # y = data


  # storage and initialization
  mu_1 <- rep(0, R)
  mu_2 <- rep(0, R)
  pi <- rep(0.5, R)

  # sample size
  n <- length(y)

  # sampling
  for (j in 2:R) {
    # sample z
    # sample mu_1



    # sample mu_2



    # sample pi



  }

 return(list(mu_1 = mu_1, mu_2 = mu_2, pi = pi))

}
```