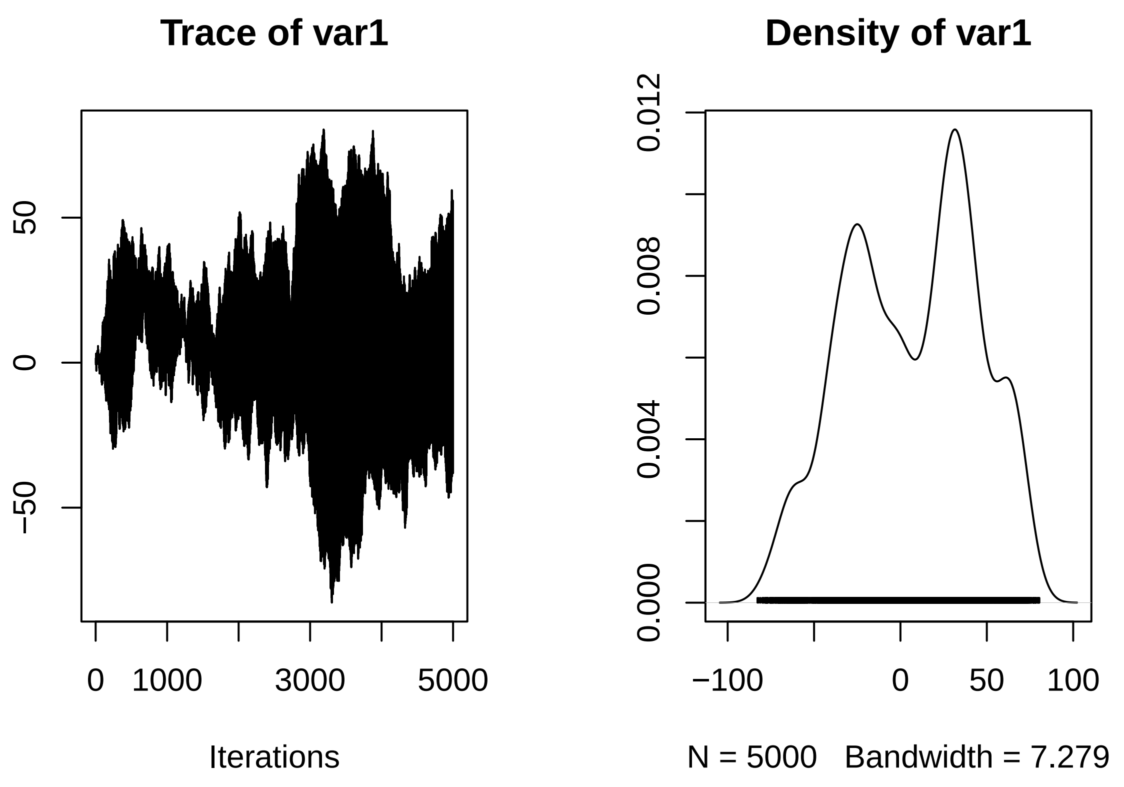
```r
# There are three parameters of interest: theta1, theta2, theta3
# Define the likelihood function
likelihood <- function(theta) {
  return(dnorm(theta[1], mean = 0, sd = 1) *
         dnorm(theta[2], mean = theta[1], sd = 1) *
         dnorm(theta[3], mean = theta[2], sd = 1))
}

# Define the prior distribution
prior <- function(theta) {
  return(dnorm(theta[1], mean = 0, sd = 1) *
         dnorm(theta[2], mean = 0, sd = 1) *
         dnorm(theta[3], mean = 0, sd = 1))
}

# Define the posterior distribution
posterior <- function(theta) {
  return(likelihood(theta) * prior(theta))
}
```

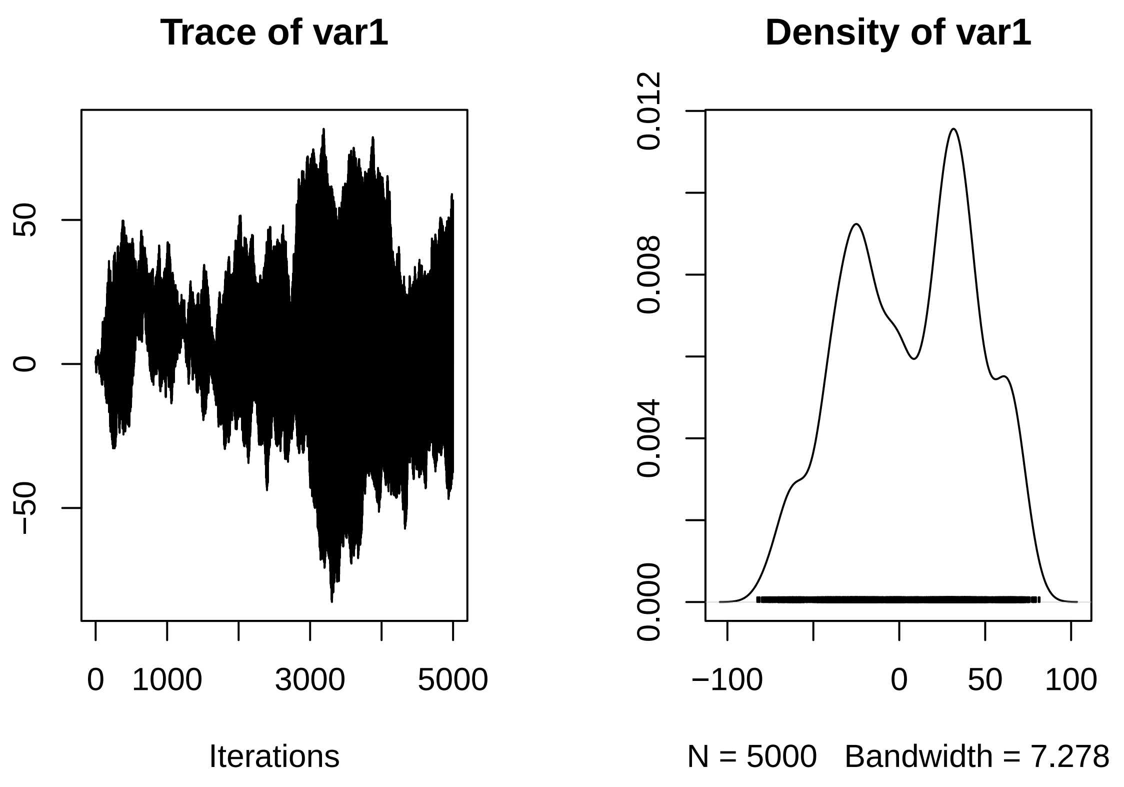
```r
# Define the Gibbs sampler
gibbs_sampler <- function(theta_0, iter) {
  theta1 <- theta_0[1]
  theta2 <- theta_0[2]
  theta3 <- theta_0[3]
  samples <- matrix(numeric(iter * 3), ncol = 3)

  for (i in 1:iter) {
    theta1 <- rnorm(1, mean = theta2, sd = 1)
    theta2 <- rnorm(1, mean = theta3, sd = 1)
    theta3 <- rnorm(1, mean = theta1, sd = 1)
    samples[i, ] <- c(theta1, theta2, theta3)
  }

  return(samples)
}
```

```r
library(coda)

# Run the Gibbs sampler with two different starting points
# Note the second starting choice is a poor choice
#the chain does not converge within 5000 interations
set.seed(123)
samples1 <- as.mcmc(gibbs_sampler(theta_0 = c(0, 0, 0), iter = 5000))
set.seed(123)
samples2 <- as.mcmc(gibbs_sampler(theta_0 = c(6,7,8), iter = 5000))

# Combine the results and check for convergence
samples_list <- list(samples1, samples2)
```

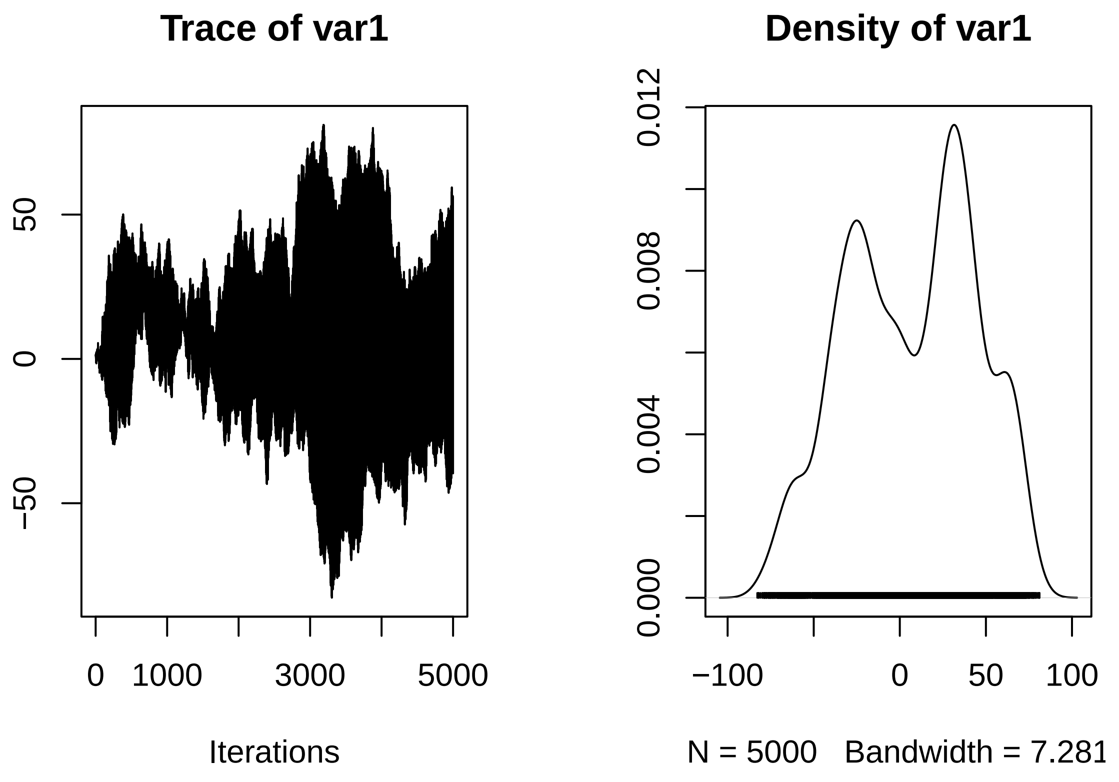
```r
plot(samples1[,1])
```



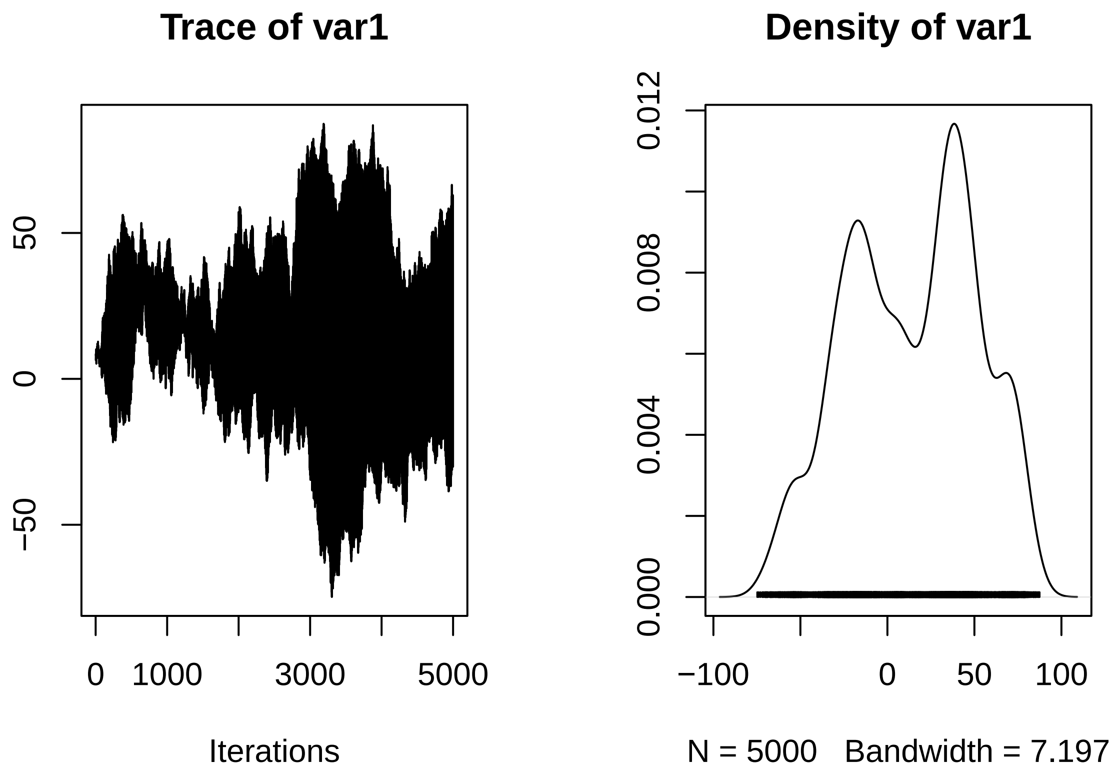
```r
plot(samples1[,2])
```



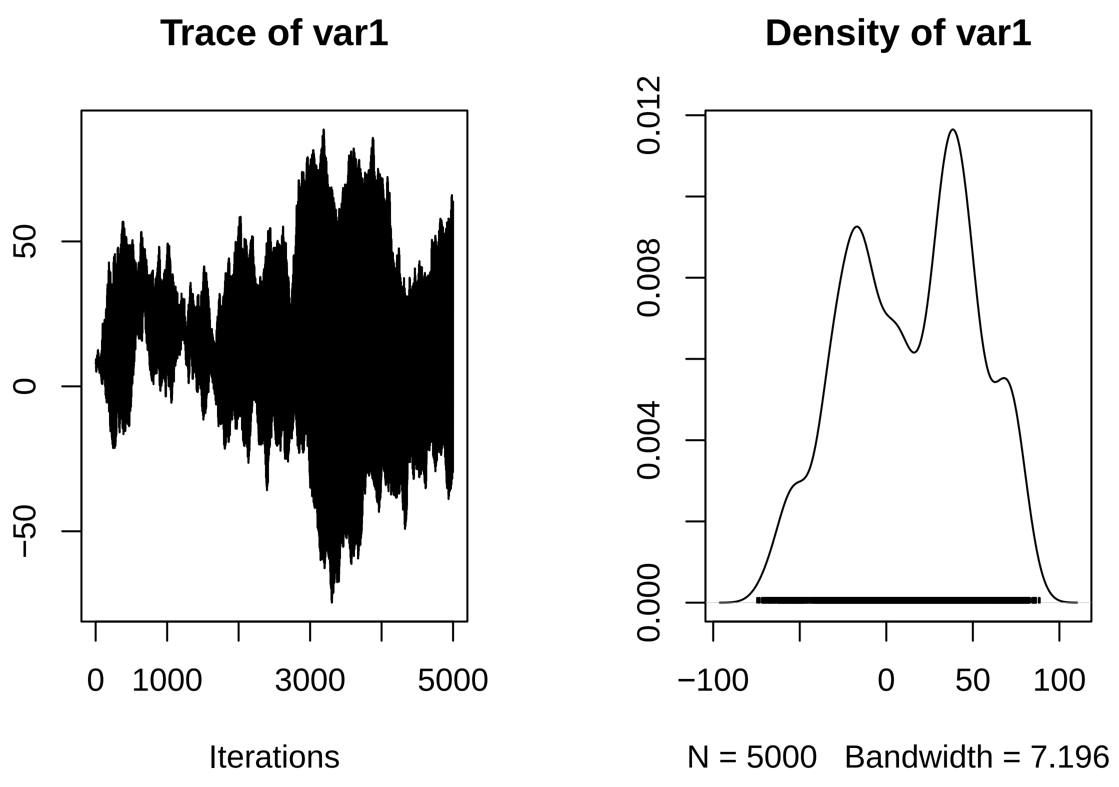
```r
plot(samples1[,3])
```



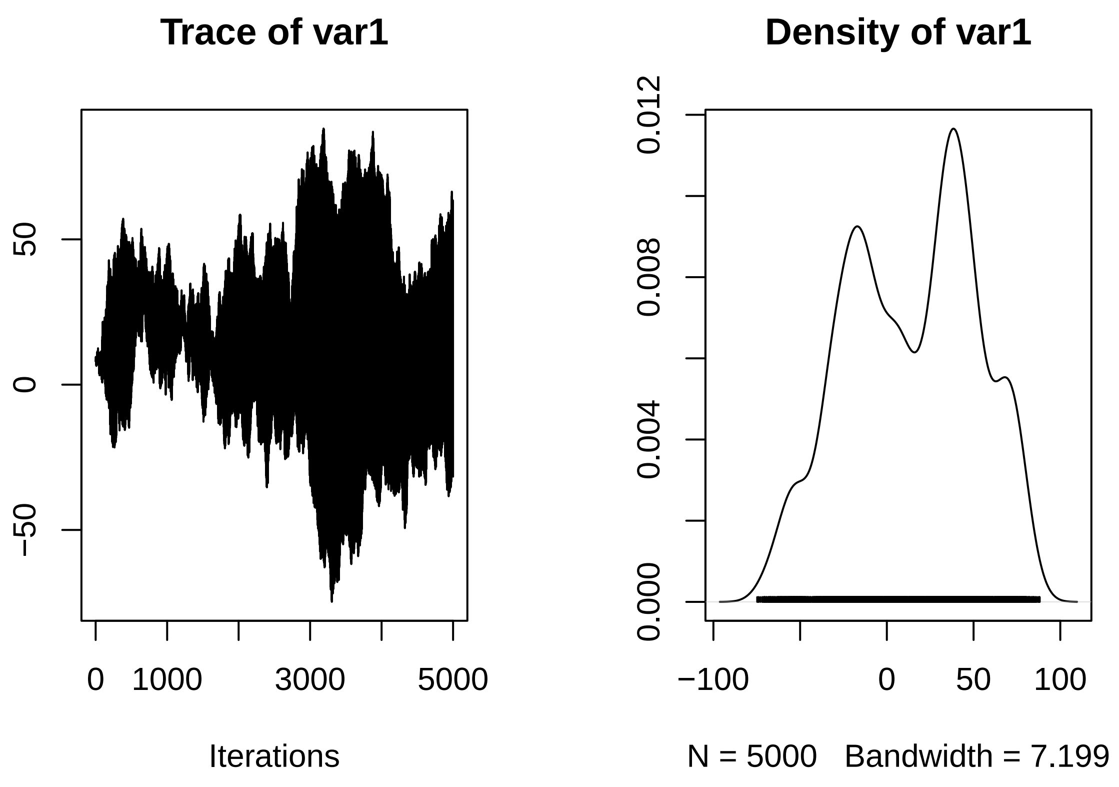
```
plot(samples2[,1])
```



```
plot(samples2[,2])
```



```
plot(samples2[,3])
```



```
#Z-score for a test of equality that compares the means of the first and last parts of each chain
geweke.diag(samples1[,])
```

```
##
## Fraction in 1st window = 0.1
## Fraction in 2nd window = 0.5
##
##   var1   var2   var3
## 0.7032 0.6947 0.7195
```

```
geweke.diag(samples2[,])
```

```
##
## Fraction in 1st window = 0.1
## Fraction in 2nd window = 0.5
##
##   var1   var2   var3
## 0.7020 0.6939 0.7184
```

```
#Gelman-Rubin diagnostic
(diagnostics <- gelman.diag(samples_list))  #need at least two chains to check for convergence
```

```
## Potential scale reduction factors:
##
##      Point est. Upper C.I.
## [1,]       1.01       1.05
## [2,]       1.01       1.05
## [3,]       1.01       1.05
##
## Multivariate psrf
##
## 1.24
```

```
#This will produce the potential scale reduction factor (PSRF) for each parameter,
#allowing you to check for convergence of the Metropolis-Hastings algorithm.
#If the PSRF is substantially greater than 1.1 for any parameter, it suggests
#that the chain has not converged.

#Effective Sample Size
effectiveSize(samples_list)
```

```
##     var1     var2     var3
## 420.6048 412.0096 379.6192
```

```
# Part A

##Using the gelman-rubin we see the value of point estimation and upper CI is approximately 1 which ind
## Using the Geweke, since the z-score is less than 2 we can say that the chains have converged.
## The effective sample size, we see that the chain has converged a bit as it is > 100
```

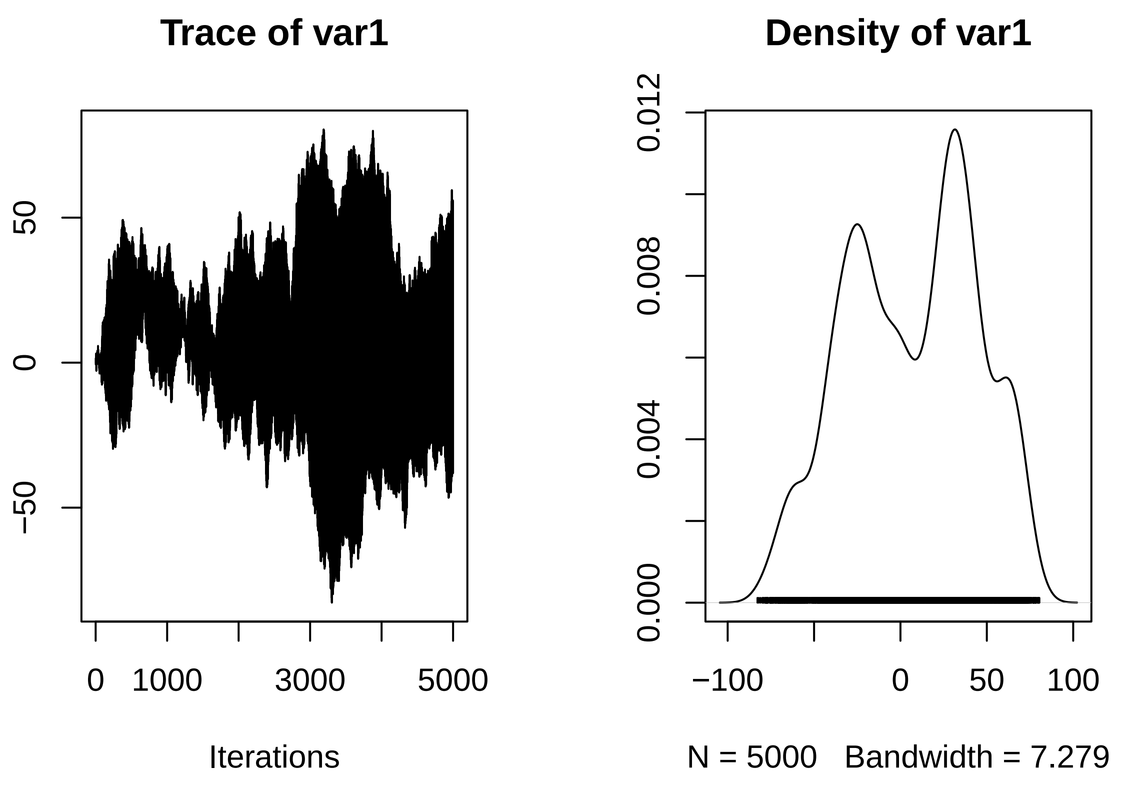
```
# There are three parameters of interest: theta1, theta2, theta3
# Define the likelihood function
likelihood <- function(theta) {
  return(dnorm(theta[1], mean = 0, sd = 1) *
```

```r
         dnorm(theta[2], mean = theta[1], sd = 1) *
         dnorm(theta[3], mean = theta[2], sd = 1))
}

# Define the prior distribution
prior <- function(theta) {
  return(dnorm(theta[1], mean = 0, sd = 1) *
         dnorm(theta[2], mean = 0, sd = 1) *
         dnorm(theta[3], mean = 0, sd = 1))
}

# Define the posterior distribution
posterior <- function(theta) {
  return(likelihood(theta) * prior(theta))
}
```

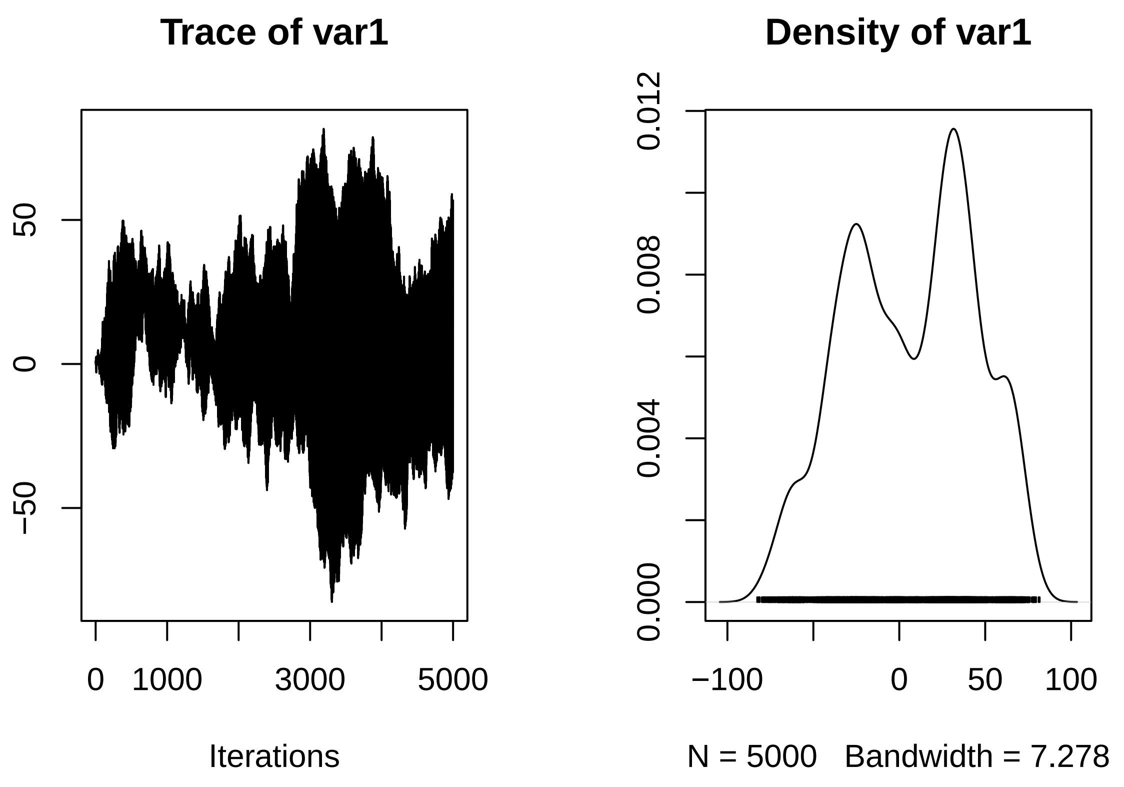
```r
# Define the Gibbs sampler
gibbs_sampler <- function(theta_0, iter) {
  theta1 <- theta_0[1]
  theta2 <- theta_0[2]
  theta3 <- theta_0[3]
  samples <- matrix(numeric(iter * 3), ncol = 3)

  for (i in 1:iter) {
    theta1 <- rnorm(1, mean = theta2, sd = 1)
    theta2 <- rnorm(1, mean = theta3, sd = 1)
    theta3 <- rnorm(1, mean = theta1, sd = 1)
    samples[i, ] <- c(theta1, theta2, theta3)
  }

  return(samples)
}
```

```r
library(coda)

# Run the Gibbs sampler with two different starting points
# Note the second starting choice is a poor choice
#the chain does not converge within 5000 interations
set.seed(123)
samples1 <- as.mcmc(gibbs_sampler(theta_0 = c(0, 0, 0), iter = 5000))
set.seed(123)
samples2 <- as.mcmc(gibbs_sampler(theta_0 = c(0.3,0.3,0.3), iter = 5000))

# Combine the results and check for convergence
samples_list <- list(samples1, samples2)
```

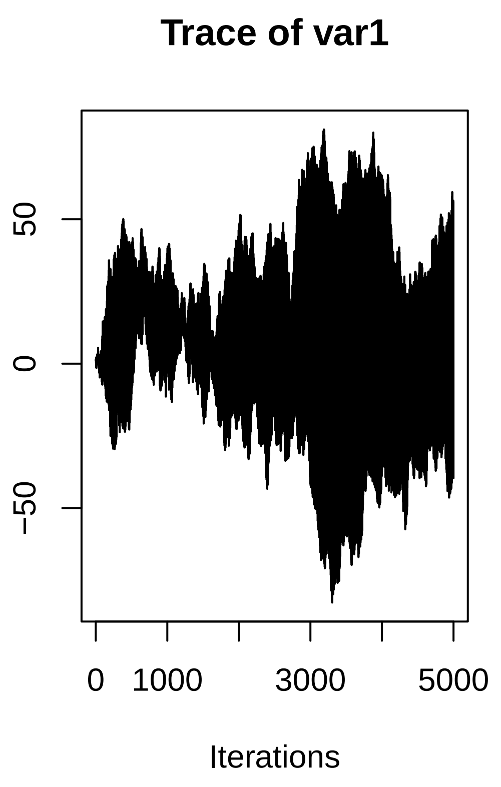
```r
plot(samples1[,1])
```

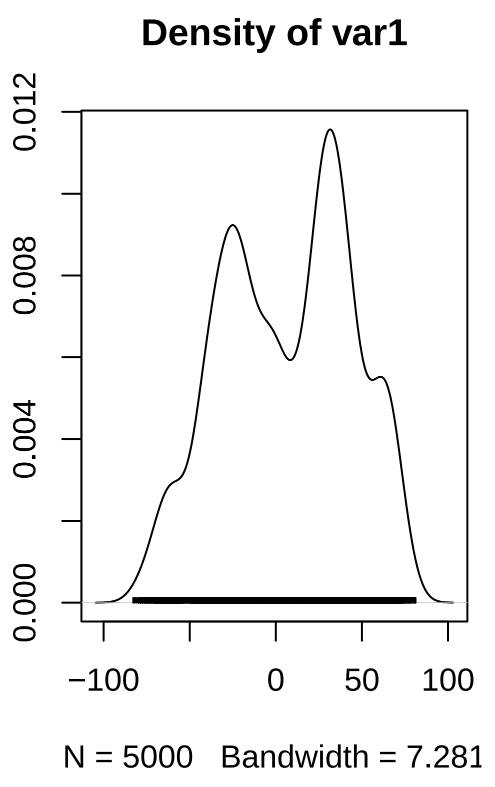


```r
plot(samples1[,2])
```

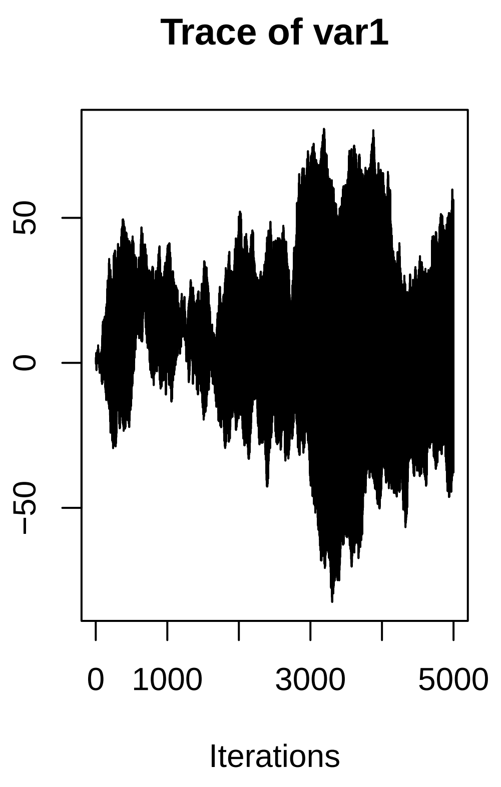


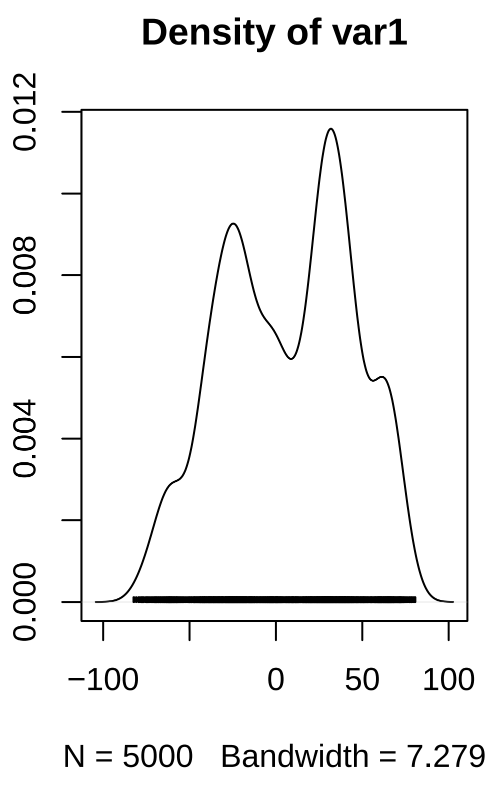
```r
plot(samples1[,3])
```



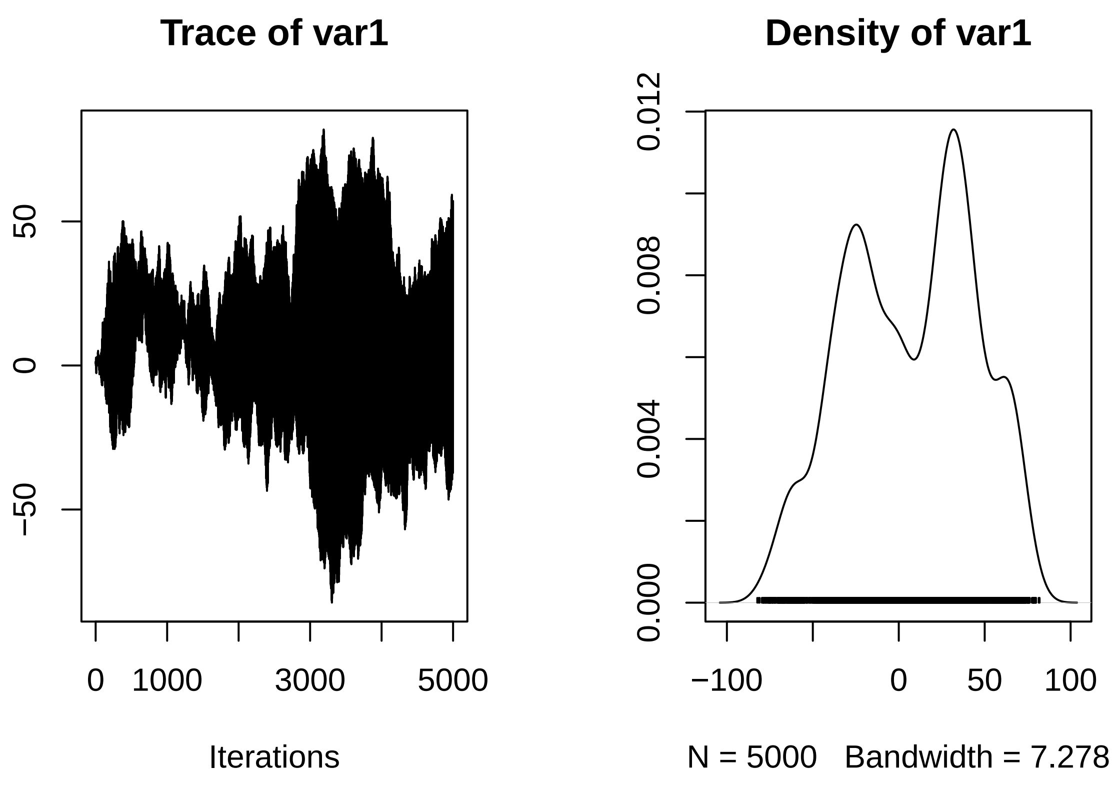


```
plot(samples2[,1])
```

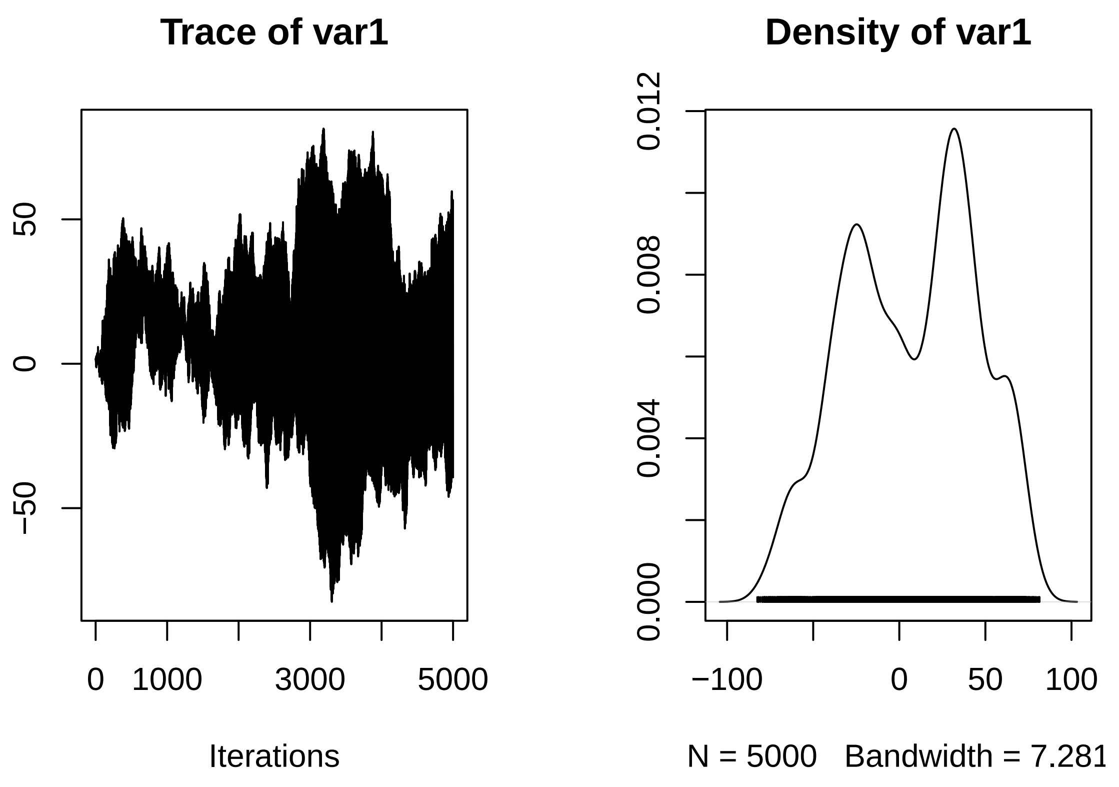




```
plot(samples2[,2])
```



```
plot(samples2[,3])
```



```
#Z-score for a test of equality that compares the means of the first and last parts of each chain
geweke.diag(samples1[,])
```

```
##
## Fraction in 1st window = 0.1
## Fraction in 2nd window = 0.5
##
##   var1   var2   var3
## 0.7032 0.6947 0.7195
```

```
geweke.diag(samples2[,])
```

```
##
## Fraction in 1st window = 0.1
## Fraction in 2nd window = 0.5
##
##   var1   var2   var3
## 0.7032 0.6947 0.7195
```

```
#Gelman-Rubin diagnostic
(diagnostics <- gelman.diag(samples_list))  #need at least two chains to check for convergence
```

```
## Potential scale reduction factors:
##
##      Point est. Upper C.I.
## [1,]          1          1
## [2,]          1          1
## [3,]          1          1
##
## Multivariate psrf
##
## 1
```

```
#This will produce the potential scale reduction factor (PSRF) for each parameter,
#allowing you to check for convergence of the Metropolis-Hastings algorithm.
#If the PSRF is substantially greater than 1.1 for any parameter, it suggests
#that the chain has not converged.

#Effective Sample Size
effectiveSize(samples_list)
```

```
##     var1     var2     var3
## 621.5855 610.7937 569.6906
```

```
# Part B
## On reducing the sample size of (0.3,0.3,0.3), the chains have converged properly as seen by the valu
```