

Theoretical Exercises

1. Let θ be a random variable with density $p(\theta)$ —the target density—and θ^* be a random variable with density $g(\theta^*)$ —the proposal density. Further, let M be some finite, nonnegative value such that $p(\theta) \leq Mg(\theta)$. Prove that the rejection sampler (or accept/reject algorithm) results in samples from the target density. Hint: It is useful to begin by working with the CDF of θ , $P(\theta \leq a)$, and noting that conditional on the acceptance rule, $P(\theta \leq a) = P(\theta^* \leq a | \text{accept } \theta^*)$.

Let $\theta \sim p(\theta)$ and $\theta^* \sim g(\theta^*)$, where $p(\theta)$ and $g(\theta^*)$ have common support with $M = \sup_{\theta} \frac{p(\theta)}{Mg(\theta^*)} < \infty$

To generate a random variable $\theta \sim p(\theta)$:

- a. Generate $U \sim \text{Unif}(0, 1)$ and $\theta^* \sim g(\theta^*)$, independent.
- b. If $U < \frac{p(\theta^*)}{Mg(\theta^*)}$, set $\theta = \theta^*$; otherwise, return to step (a)

The generated random variable θ has cdf $P(\theta \leq a)$

$$\begin{aligned}
 P(\theta \leq a) &= P(\theta^* \leq a | \text{accept } \theta^*) \\
 &= P\left(\theta^* \leq a \mid U < \frac{p(\theta^*)}{Mg(\theta^*)}\right) \\
 &= \frac{P\left(\theta^* \leq a, U < \frac{p(\theta^*)}{Mg(\theta^*)}\right)}{P\left(U < \frac{p(\theta^*)}{Mg(\theta^*)}\right)} \\
 &= \frac{\int_{-\infty}^a \int_0^{\frac{p(\theta^*)}{Mg(\theta^*)}} d\theta \, g(\theta^*) \, d\theta^*}{\int_{-\infty}^{\infty} \int_0^{\frac{p(\theta^*)}{Mg(\theta^*)}} d\theta \, g(\theta^*) \, d\theta^*} \\
 &= \int_{-\infty}^a p(\theta^*) \, d\theta^*
 \end{aligned}$$

which is the desired cdf.

2. Let $y_i \stackrel{iid}{\sim} N(\mu, \alpha\sigma^2)$ and place the following priors on the model components: $\pi(\mu) \propto 1$, $\alpha \sim IG(a, b)$, and $\sigma^2 \sim IG(c, d)$ for constants a, b, c , and d . This is an example of a parameter expanded normal model. Find the posterior and determine the full conditionals for the model. Using these, write the steps necessary to use a Gibbs sampler to draw samples from μ , α , and σ^2 .

$$\mathcal{L}(y_i|\mu, \sigma^2, \alpha) \propto (\alpha\sigma^2)^{-\frac{n}{2}} \exp \left[-\frac{1}{2\alpha\sigma^2} \sum_i (y_i - \mu)^2 \right]$$

$$\begin{aligned} p(\mu, \alpha, \sigma^2|y) &\propto (\alpha\sigma^2)^{-\frac{n}{2}} \exp \left[-\frac{1}{2\alpha\sigma^2} \sum_i (y_i - \mu)^2 \right] \frac{b^a}{\Gamma(a)} \alpha^{-(a+1)} \exp \left[-\frac{b}{\alpha} \right] \frac{d^c}{\Gamma(c)} (\sigma^2)^{-(c+1)} \exp \left[-\frac{d}{\sigma^2} \right] \\ &\propto (\alpha\sigma^2)^{-\frac{n}{2}} \exp \left[-\frac{1}{2\alpha\sigma^2} \sum_i (y_i - \mu)^2 \right] \alpha^{-(a+1)} \exp \left[-\frac{b}{\alpha} \right] (\sigma^2)^{-(c+1)} \exp \left[-\frac{d}{\sigma^2} \right] \end{aligned}$$

$$\begin{aligned} p(\alpha|\mu, \sigma^2, y) &\propto \alpha^{-\frac{n}{2}} \exp \left[-\frac{1}{2\alpha\sigma^2} \sum_i (y_i - \mu)^2 \right] \alpha^{-(a+1)} \exp \left[-\frac{b}{\alpha} \right] \\ &\propto \alpha^{-(\frac{n}{2}+a+1)} \exp \left[-\frac{1}{\alpha} \left(b + \frac{1}{2\sigma^2} \sum_i (y_i - \mu)^2 \right) \right] \end{aligned}$$

$$\alpha|\mu, \sigma^2, y \sim IG \left(\frac{n}{2} + a, b + \frac{1}{2\sigma^2} \sum_i (y_i - \mu)^2 \right)$$

$$\begin{aligned} p(\sigma^2|\mu, \alpha, y) &\propto (\sigma^2)^{-\frac{n}{2}} \exp \left[-\frac{1}{2\alpha\sigma^2} \sum_i (y_i - \mu)^2 \right] (\sigma^2)^{-(c+1)} \exp \left[-\frac{d}{\sigma^2} \right] \\ &\propto (\sigma^2)^{-(\frac{n}{2}+c+1)} \exp \left[-\frac{1}{\sigma^2} \left(d + \frac{1}{2\alpha} \sum_i (y_i - \mu)^2 \right) \right] \end{aligned}$$

$$\sigma^2|\mu, \alpha, y \sim IG \left(\frac{n}{2} + c, d + \frac{1}{2\alpha} \sum_i (y_i - \mu)^2 \right)$$

$$\begin{aligned}
p(\mu|\sigma^2, \alpha^2, y) &\propto \exp \left[-\frac{1}{2\alpha\sigma^2} \sum_i (y_i - \mu)^2 \right] \\
&\propto \exp \left[-\frac{1}{2\frac{\alpha\sigma^2}{n}} \sum_i (\mu - \bar{y})^2 \right] \\
\mu|\sigma^2, \alpha, y &\sim N \left(\bar{y}, \frac{\alpha\sigma^2}{n} \right)
\end{aligned}$$

3. Steps necessary to use a Gibbs sampler (b^* some constant):

$$\begin{aligned}
\alpha^{(b)} &\sim IG \left(\frac{n}{2} + a, b^* + \frac{1}{2(\sigma^2)^{(b-1)}} \sum_i \left(y_i - \mu^{(b-1)} \right)^2 \right) \\
\sigma^{2^{(b)}} &\sim IG \left(\frac{n}{2} + c, d + \frac{1}{2\alpha^{(b)}} \sum_i \left(y_i - \mu^{(b-1)} \right)^2 \right) \\
\mu^{(b)} &\sim N \left(\bar{y}, \frac{\alpha^{(b)} (\sigma^2)^{(b)}}{n} \right)
\end{aligned}$$

3. Let $x_i \stackrel{iid}{\sim} N(\mu, \sigma^2)$ and place the flat prior on μ , $\pi(\mu) \propto 1$. Next assume $\sigma^2 \sim IG(v/2, v/\alpha)$ and $\alpha \sim IG(1/2, 1/A^2)$ for fixed values v and A . When $v = 1$, this prior and hyperprior combination are equivalent to placing a Half-Cauchy prior with scale A on σ^2 —a useful prior for variances, particularly when they are near zero. Express the posterior and determine the full conditionals for the model. Then write out the steps for the Gibbs sampler to draw samples from μ , σ^2 , and α .

$$\mathcal{L}(x_i|\mu, \sigma^2) \propto (\sigma^2)^{-\frac{n}{2}} \exp \left[\frac{1}{2\sigma^2} \sum_i (x_i - \mu)^2 \right]$$

$$\begin{aligned} p(\mu, \sigma^2, \alpha|x) &\propto (\sigma^2)^{-\frac{n}{2}} \exp \left[-\frac{1}{2\sigma^2} \sum_i (x_i - \mu)^2 \right] \frac{\left(\frac{v}{\alpha}\right)^{\frac{v}{2}}}{\Gamma\left(\frac{v}{2}\right)} (\sigma^2)^{-\left(\frac{v}{2}+1\right)} \exp \left[-\frac{v}{\sigma^2} \right] \frac{\frac{1}{A^2}}{\Gamma\left(\frac{1}{2}\right)} \alpha^{-\left(\frac{1}{2}+1\right)} \exp \left[-\frac{1}{A^2\alpha} \right] \\ &\propto (\sigma^2)^{-\left(\frac{n+v}{2}+1\right)} \exp \left[-\frac{1}{2\sigma^2} \sum_i (x_i - \mu)^2 \right] \left(\frac{v}{\alpha}\right)^{\frac{v}{2}} \exp \left[-\frac{v}{\alpha\sigma^2} \right] \alpha^{-1.5} \exp \left[-\frac{1}{A^2\alpha} \right] \end{aligned}$$

$$\begin{aligned} p(\alpha|\mu, \sigma^2, y) &\propto \left(\frac{v}{\alpha}\right)^{\frac{v}{2}} \exp \left[-\frac{v}{\alpha\sigma^2} \right] \alpha^{-1.5} \exp \left[-\frac{1}{A^2\alpha} \right] \\ &\propto \alpha^{-\left(\frac{v+1}{2}+1\right)} \exp \left[-\frac{1}{\alpha} \left(\frac{v}{\sigma^2} + \frac{1}{A^2} \right) \right] \end{aligned}$$

$$\alpha|\mu, \sigma^2, y \sim IG\left(\frac{v+1}{2}, \frac{v}{\sigma^2} + \frac{1}{A^2}\right)$$

$$\begin{aligned} p(\sigma^2|\mu, \alpha, y) &\propto (\sigma^2)^{-\left(\frac{n+v}{2}+1\right)} \exp \left[-\frac{1}{2\sigma^2} \sum_i (x_i - \mu)^2 \right] \exp \left[-\frac{v}{\alpha\sigma^2} \right] \\ &\propto (\sigma^2)^{-\left(\frac{n+v}{2}+1\right)} \exp \left[-\frac{1}{\sigma^2} \left(\frac{v}{\alpha} + \frac{1}{2} \sum_i (x_i - \mu)^2 \right) \right] \end{aligned}$$

$$\sigma^2|\mu, \alpha, y \sim IG\left(\frac{n+v}{2}, \frac{v}{\alpha} + \frac{1}{2} \sum_i (x_i - \mu)^2\right)$$

$$\begin{aligned}
p(\mu|\sigma^2, \alpha, y) &\propto \exp \left[-\frac{1}{2\sigma^2} \sum_i (x_i - \mu)^2 \right] \\
&\propto \exp \left[-\frac{1}{2\frac{\sigma^2}{n}} \sum_i (\mu - \bar{x})^2 \right] \\
\mu|\sigma^2, \alpha, y &\sim N \left(\bar{x}, \frac{\sigma^2}{n} \right)
\end{aligned}$$

Steps necessary to use a Gibbs sampler:

$$\begin{aligned}
\alpha^{(b)} &\sim IG \left(\frac{v+1}{2}, \frac{v}{(\sigma^2)^{(b-1)}} + \frac{1}{A^2} \right) \\
(\sigma^2)^{(b)} &\sim IG \left(\frac{n+v}{2}, \frac{v}{\alpha^{(b)}} + \frac{1}{2} \sum_i \left(x_i - \mu^{(b-1)} \right)^2 \right) \\
\mu^{(b)} &\sim N \left(\bar{x}, \frac{(\sigma^2)^{(b)}}{n} \right)
\end{aligned}$$

Computing Exercises

1. Suppose we wish to sample from the triangular distribution which has density function

$$p(\theta) = \begin{cases} 0 & \theta < 0 \\ 4\theta & 0 \leq \theta < 1/2 \\ 4 - 4\theta & 1/2 \leq \theta \leq 1 \\ 0 & \theta > 1 \end{cases}.$$

This density arises as the mean of two uniformly distributed random variables. One approach to sampling from $p(\theta)$ is to use rejection sampling. For your sampler, consider two different forms of $g(\theta)$: first, let $g(\theta)$ be the standard uniform density and second, let $g(\theta)$ be the $Beta(2,2)$ density. For both proposal densities, you will need to determine the values of M such that $p(\theta) \leq Mg(\theta)$. For each $g(\theta)$, first write a rejection sampler to obtain 1000 samples from $p(\theta)$. Then compare the resulting sampled distributions as well as the acceptance rate. Is one $g(\theta)$ more efficient than the other? Explain. For each sampler, set the seed to 52918.

For the standard uniform proposal:

$$M = \sup_{\theta} \frac{P(\theta)}{g(\theta)} = \sup_{\theta} \begin{cases} 0 & \theta < 0 \\ 4\theta & 0 \leq \theta < \frac{1}{2} \\ 4 - 4\theta & \frac{1}{2} \leq \theta \leq 1 \\ 0 & \theta > 1 \end{cases} = \sup_{\theta} 4 - 4\theta = 4 - 4(0.5) = 2.$$

For the $Beta(2,2)$ proposal:

$$M = \sup_{\theta} \frac{P(\theta)}{g(\theta)} = \sup_{\theta} \begin{cases} 0 & \theta < 0 \\ \frac{4\theta}{6\theta(1-\theta)} & 0 \leq \theta < \frac{1}{2} \\ \frac{4-4\theta}{6\theta(1-\theta)} & \frac{1}{2} \leq \theta \leq 1 \\ 0 & \theta > 1 \end{cases} = \sup_{\theta} \frac{4-4\theta}{6\theta(1-\theta)} = \frac{4-4(0.5)}{6(0.5)(1-(0.5))} = \frac{8}{6}.$$

```
fx <- function(x){ ifelse( x < 0 , 0,
  ifelse( 0 <= x & x < 0.5 , 4*x ,
  ifelse( 0.5 <= x & x <= 1, 4-4*x,
  ifelse( x < 1, 0, 0 )
))})
```

```
### U(0,1) ###
curve(fx, from = 0-.05, to = 1+.05)
M <- 2
```

```

curve(M *dunif(x, 0, 1), add=TRUE, col="gold", lwd=2)

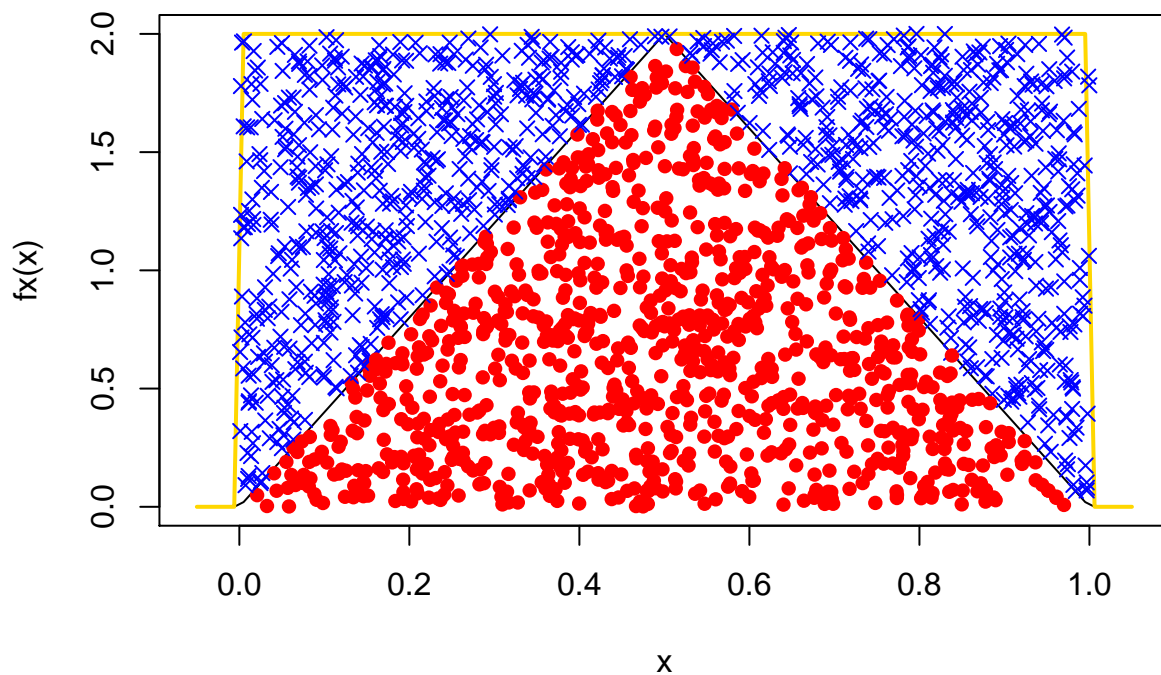
theta  <- vector(length = 1000)
arr <- NULL
t <- 1
count  <- 1

set.seed(52918)
while(t < 1000){

  tb <- runif(1)
  U  <- runif(1)

  r <- fx(tb) / (M*dunif(tb))
  if(U < r){
    theta[t] <- tb
    t <- t + 1
  }
  points( tb , M*U , col = "red" , pch = 16) } else {
  points( (tb) ,M*U , col = "blue" , pch = 4) }
  count  <- count + 1 }

```



```

t/count

## [1] 0.4933399
### Beta(2,2) ###
curve(fx, from = 0-.05, to = 1+.05 )

```

```

M <- fx(.5) / ( (0.5)*(1-0.5) * (6) )
curve(M *dbeta(x, 2, 2), add=T, col="darkblue", lwd=2)

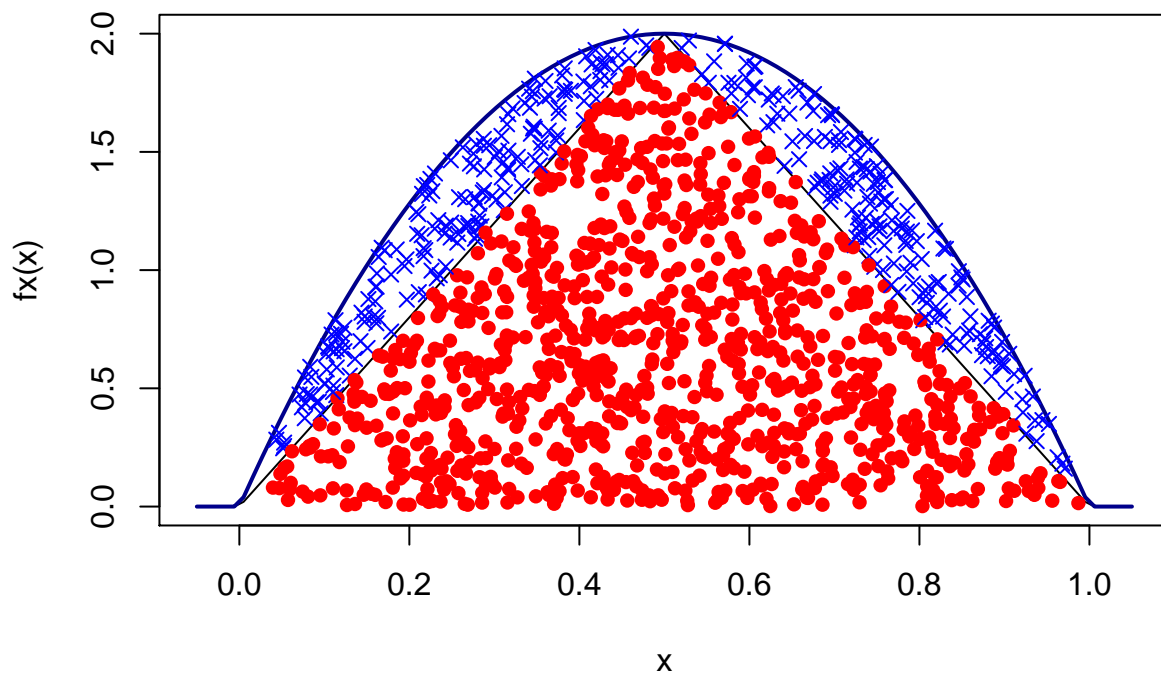
theta <- vector(length = 1000)
arr <- NULL
t <- 1
count <- 1

set.seed(52918)
while(t < 1000){

  tb <- rbeta(1,2,2)
  U <- runif(1)

  r <- fx(tb) / (M*dbeta(tb,2,2))
  if(U < r){
    theta[t] <- tb
    t <- t + 1
  }
  points( tb , M*U*dbeta(tb,2,2) , col = "red" , pch = 16) } else {
  points( (tb) ,M*U*dbeta(tb,2,2) , col = "blue" , pch = 4) }
  count <- count + 1 }

```



```
t/count
```

```
## [1] 0.7336757
```


The acceptance rate for the $Beta(2, 2)$ is about 24 percentage points higher and much more efficient than the standard uniform. We can see that the “envelope” that the $Beta(2, 2)$ forms around the target is much tighter than the one formed around the target from the standard uniform.

Analysis Exercises

Data: for the following exercise, use the dataset `hersreg.txt` from the course data file which we’ve used several times class. As a reminder, the data comes from the Heart and Estrogen-progestin Replacement Study (HERS) which was a randomized, blinded trial in post menopausal women with coronary artery disease. The study aimed to test the hypothesis that, among women with an intact uterus, those randomized to receive estrogen-progestin replacement therapy (premarin .625 mg daily plus medroxyprogesterone acetate 2.5 mg daily) have the same frequency of new coronary heart disease (CHD) events, such as myocardial infarction (MI) and death due to CHD, as those randomized to placebo. The variable `treatment` denotes the treatment groups, coded `asplacebo` and `hormone therapy`, while the variable `chtchol` denotes the change in total cholesterol.

1. One result we can examine as a proxy to incidence of MI and CHD death is change in total cholesterol over the study period. Each woman in the study had cholesterol levels taken from blood samples at the beginning and end of follow-up. We wish to build models to determine posterior estimates for the mean change in total cholesterol for each treatment group. For both groups, we will compare two likelihoods:
 - (a) First, build two normal models, one for each treatment group. Use the Gibbs sampler from TE.2 to draw your samples, making the priors non-informative.
 - (b) Second, build two non-central t -likelihood models assuming the scale parameter is fixed at the sample standard deviation with degrees of freedom equal to $n - 1$. Taking the non-informative flat prior, $\pi(\mu) \propto 1$, and using a Cauchy centered at the sample median as your proposal, sample from this posterior using rejection sampling (you will need to determine a value for M).

For each unknown parameter in both sets of models, generate convergence diagnostics (both visual and numerical, use Geweke for the later), find the 95% credible intervals, and the posterior medians. Also, under each likelihood, test to see if there is a difference in means between treatment groups. Compare your results. Use $B = 10000$ posterior samples and set the seed to 32717 for the normal models and 50814 for the t models.

Normal Model for the Placebo Group

```
setwd("G:\\math\\640")
require(mvtnorm, quietly = T)
suppressMessages(require(MCMCpack, quietly = T))
require(mcmcplots, quietly = T)
suppressMessages(require(invgamma, quietly = T))
require(ggplot2, quietly = T)
source("multipilot.R")

hers <- read.table('hersreg.txt', header = T)
colnames(hers)[1] <- "y"

hers0 <- hers[ which(hers[,2] == 0), 1:2]
```

```

B <- 10000

ss <- Alpha <- mu <- rep(NA,B)
ss[1] <- Alpha[1] <- mu[1] <- 1
v <- 1
n <- nrow(hers0)

a <- b <- cc <- d <- .0001

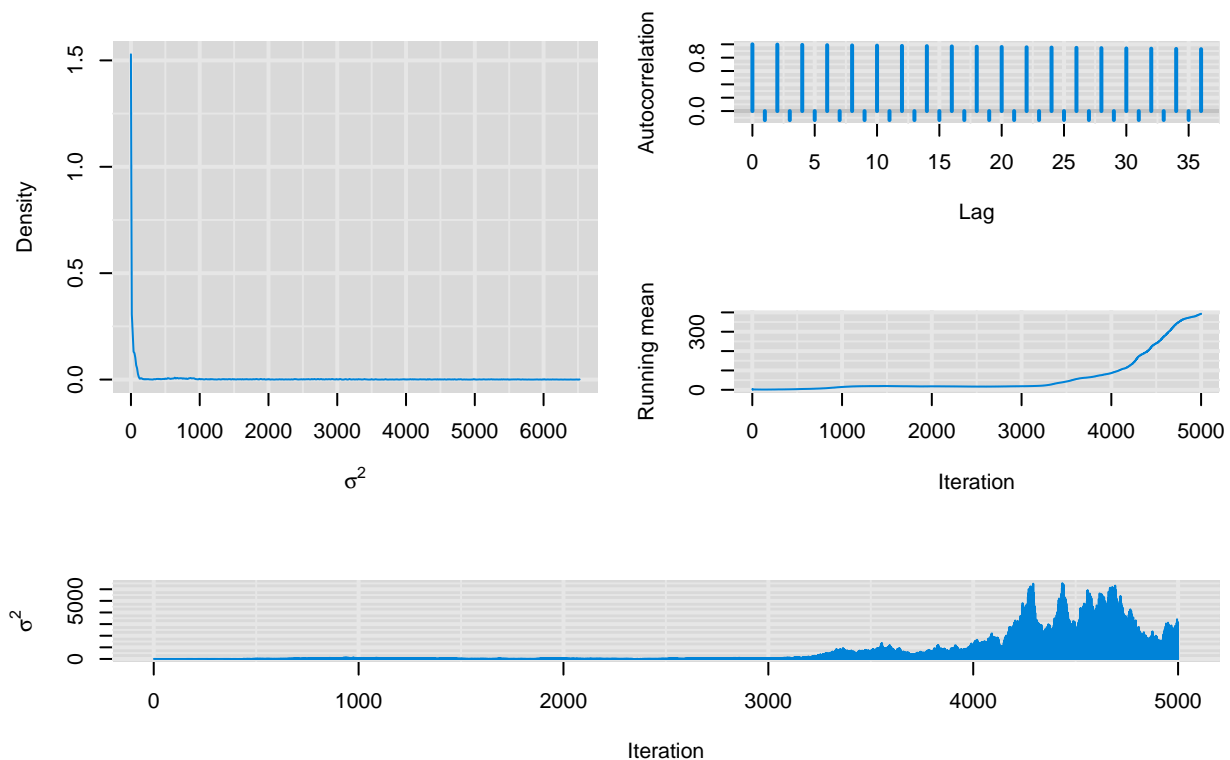
set.seed(32717)
for (t in 2:B) {
  Alpha[t] <- rinvgamma(1, (n)/2 + a, b + (1/2 * ss[t-1]) * sum( (hers0$y - mu[t-1])^2 ))
  mu[t] <- rnorm(1, mean(hers0$y), sqrt(Alpha[t-1]*ss[t-1] / n))
  ss[t] <- rinvgamma(1, (n/2) + cc, d + (1/(2*Alpha[t])) * sum( (hers0$y - mu[t])^2 ))
}

# Burn in
Alpha <- matrix( tail(Alpha,B/2), ncol = 1); colnames(Alpha) <- "alpha"
mu <- mu0a <- matrix( tail(mu,B/2), ncol = 1); colnames(mu) <- "mu"
ss <- matrix( tail(ss,B/2), ncol = 1); colnames(ss) <- "sigma^2"

mcmcplot1( ss, greek = T )

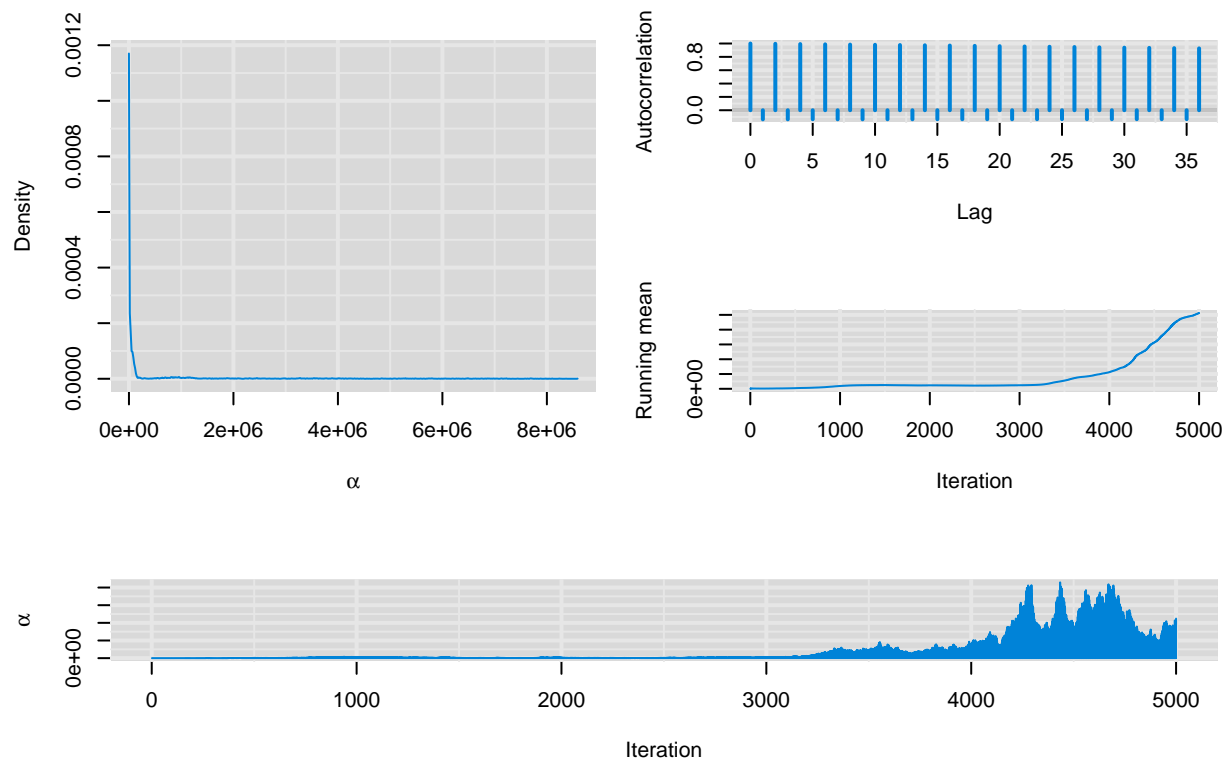
```

Diagnostics for σ^2



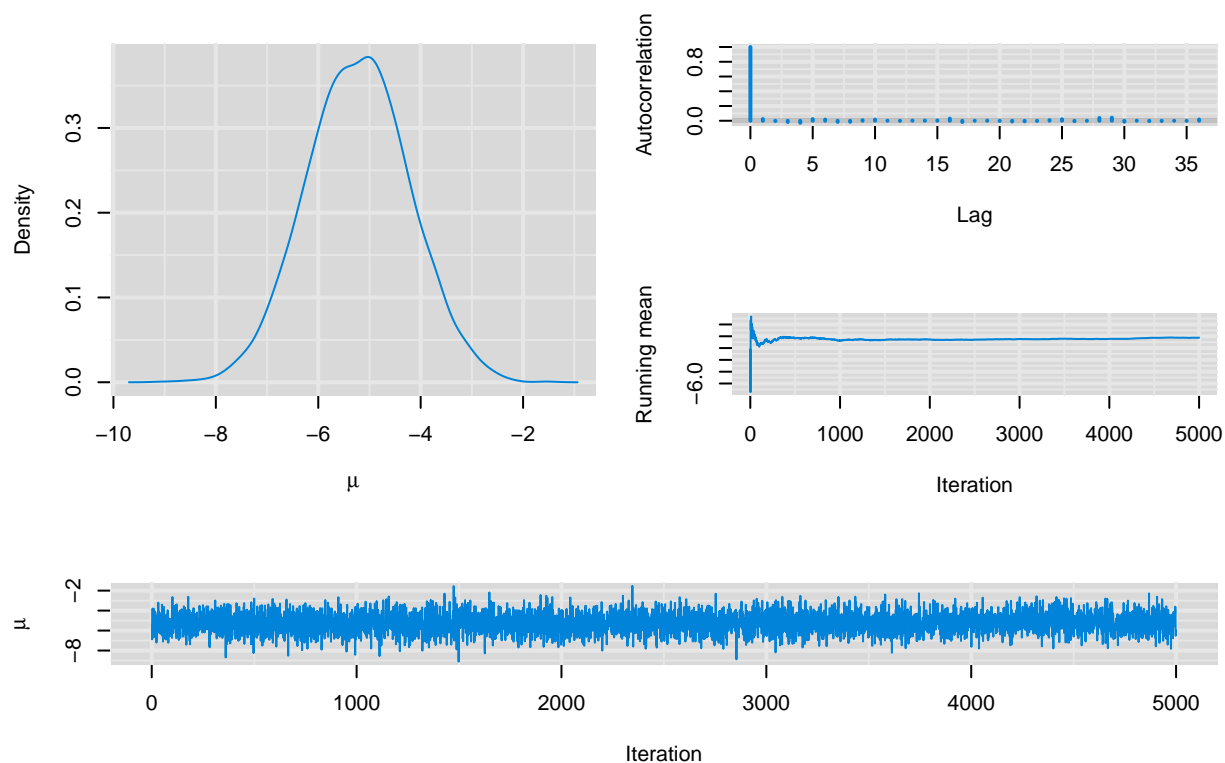
```
mcmcplot1( Alpha, greek = T )
```

Diagnostics for α



```
mcmcplot1( mu , greek = T)
```

Diagnostics for μ



```
Theta <- matrix( c(Alpha,mu,ss) , ncol = 3 ); colnames(Theta) <- c("Alpha","mu","ss")
t(apply(Theta, 2, quantile, probs = c(0.5, 0.025, 0.975)))
```

```
##           50%          2.5%          97.5%
## Alpha 1416.581119  0.3070683898  5.652895e+06
## mu      -5.218321 -7.1407320616 -3.251362e+00
## ss       1.072941  0.0002308718  4.342005e+03
```

```
geweke.diag(mcmc(Theta))
```

```
##
## Fraction in 1st window = 0.1
## Fraction in 2nd window = 0.5
##
## Alpha      mu      ss
## -1.9918 -0.8092 -1.7801
```

```
median(mu)
```

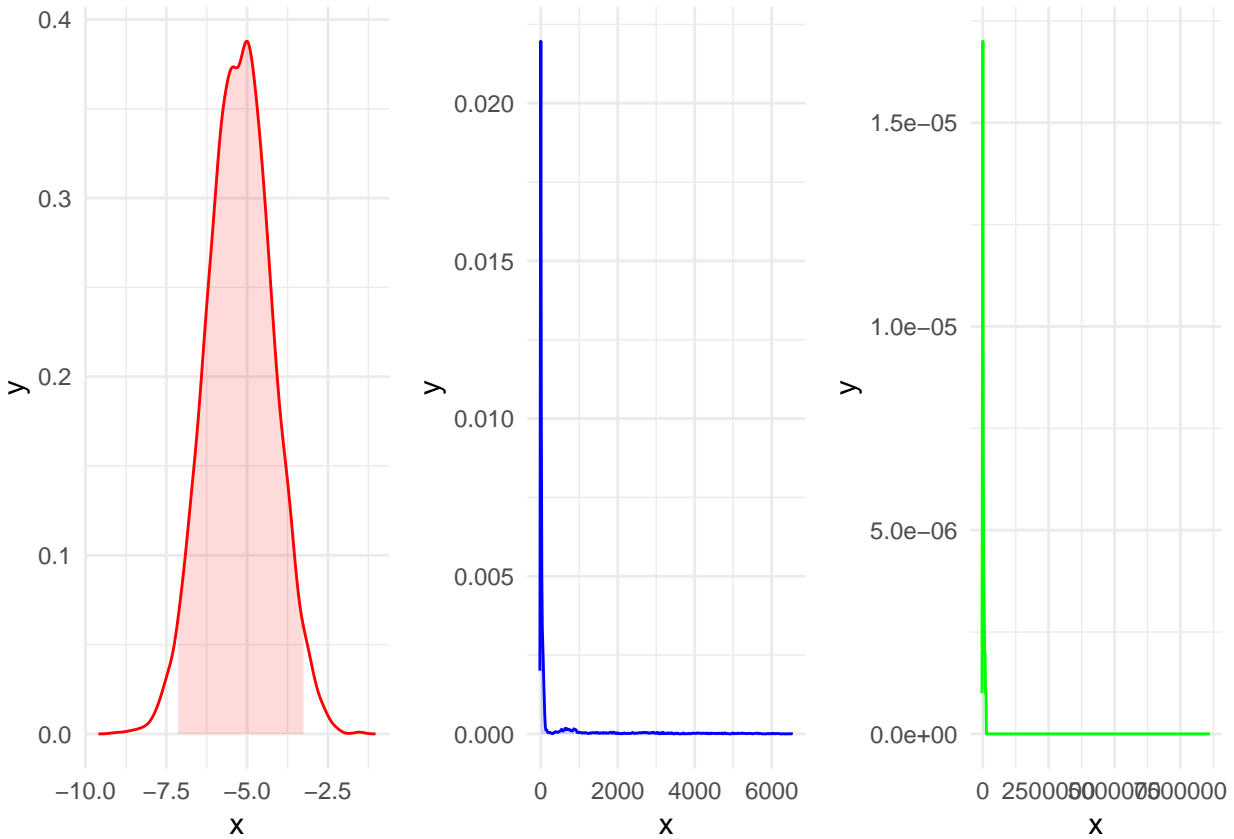
```
## [1] -5.218321
```

```
median(ss)
```

```
## [1] 1.072941
```

```
median(Alpha)
```

```
## [1] 1416.581
```



σ^2 and α are not converging well, so we select some new values for a , b , c , d and reevaluate.

```
ss <- Alpha <- mu <- rep(NA,B)
ss[1] <- Alpha[1] <- mu[1] <- 1
v <- 1
n <- nrow(hers0)

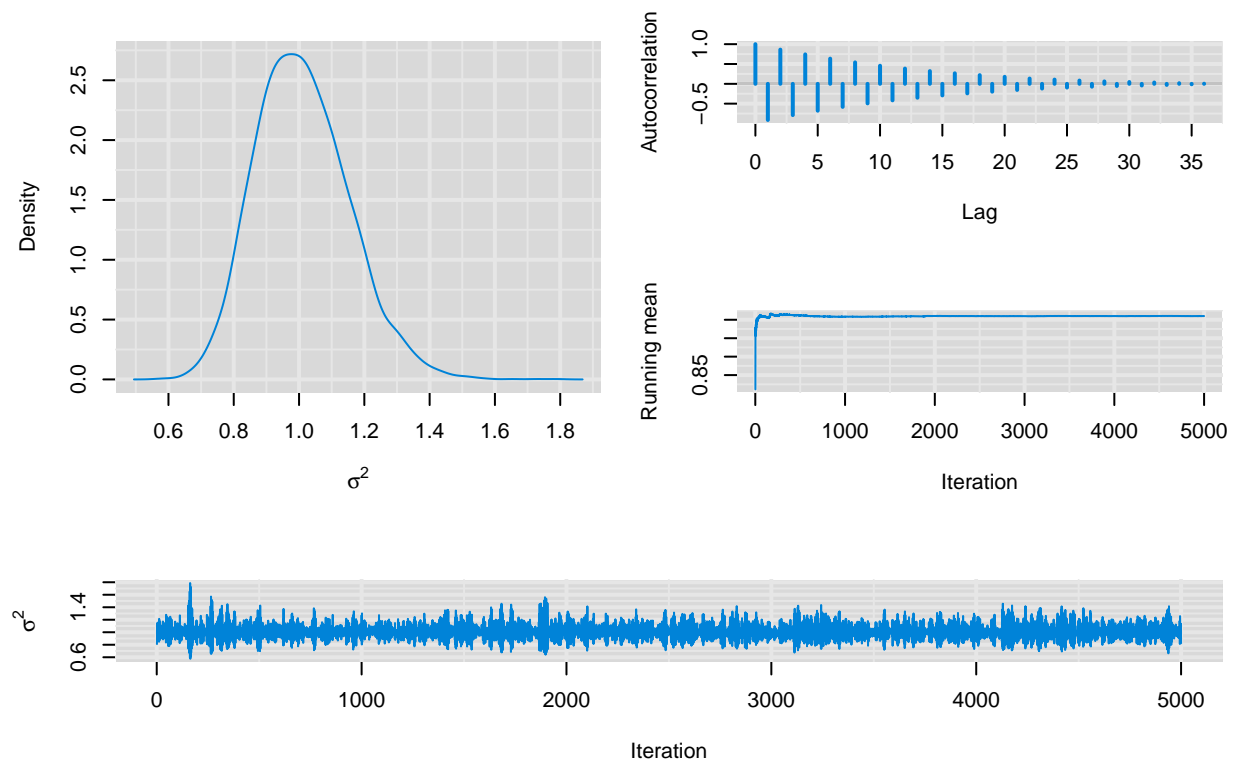
a <- b <- .1
cc <- d <- 50

set.seed(32717)
for (t in 2:B) {
  Alpha[t] <- rinvgamma(1, (n)/2 + a, b + (1/2 * ss[t-1]) * sum( (hers0$y - mu[t-1])^2 ))
  mu[t] <- rnorm(1, mean(hers0$y), sqrt(Alpha[t-1]*ss[t-1] / n))
  ss[t] <- rinvgamma(1, (n)/2 + cc, d + (1/(2*Alpha[t])) * sum( (hers0$y - mu[t])^2 ))
}

# Burn in
Alpha <- matrix( tail(Alpha,B/2), ncol = 1);colnames(Alpha) <- "alpha"
mu <- mu0b <- matrix( tail(mu,B/2), ncol = 1);colnames(mu) <- "mu"
ss <- matrix( tail(ss,B/2), ncol = 1);colnames(ss) <- "sigma^2"

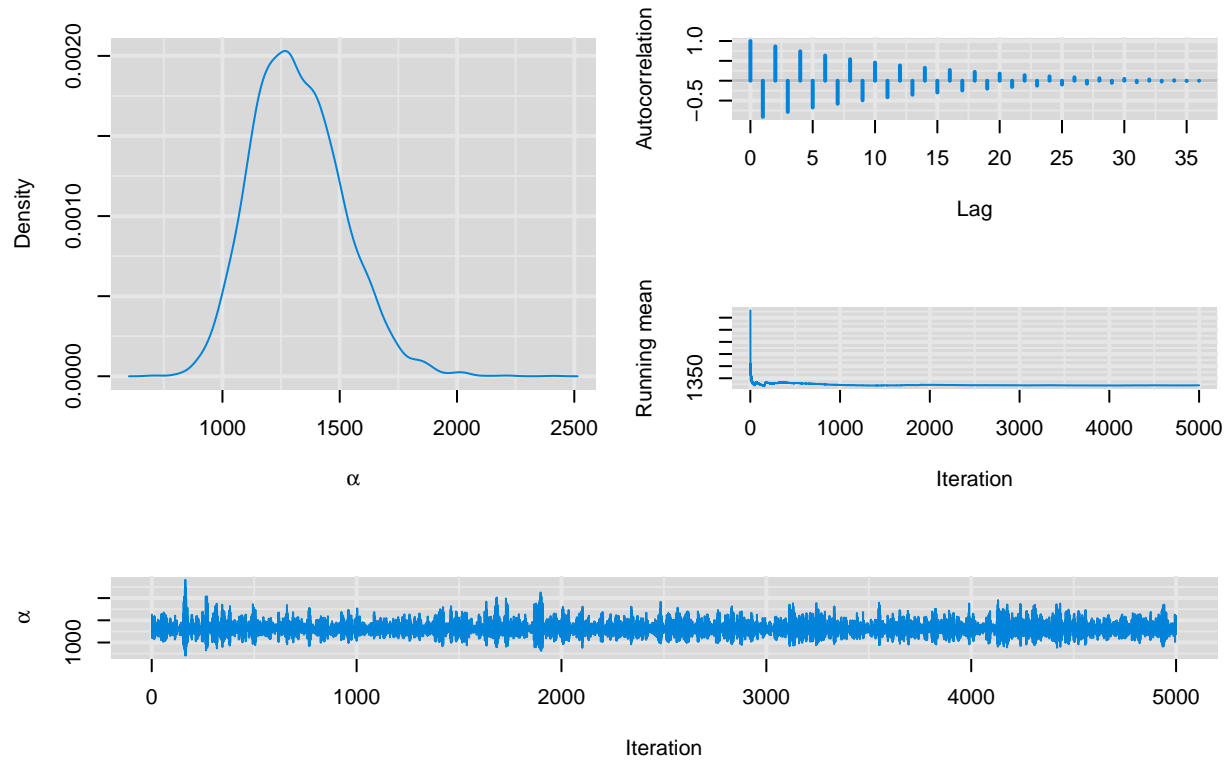
mcmcplot1( ss , greek = T )
```

Diagnostics for σ^2



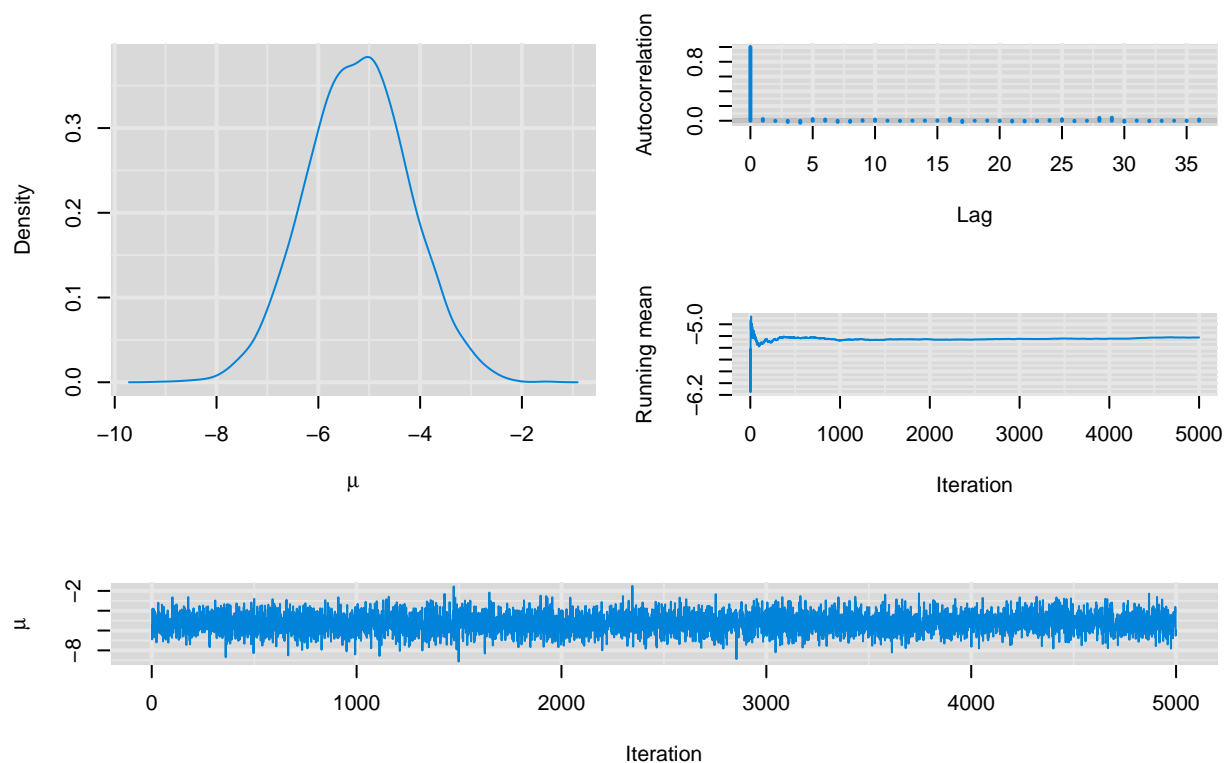
```
mcmcplot1( Alpha, greek = T )
```

Diagnostics for α



```
mcmcplot1( mu , greek = T)
```

Diagnostics for μ



```
Theta <- matrix( c(Alpha,mu,ss) , ncol = 3 ); colnames(Theta) <- c("Alpha","mu","ss")
t(apply(Theta, 2, quantile, probs = c(0.5, 0.025, 0.975)))
```

```
##           50%       2.5%      97.5%
## Alpha 1302.814949 989.2056089 1727.208783
## mu     -5.218309  -7.1456694  -3.246008
## ss      1.000116   0.7596527   1.316661
```

```
geweke.diag(mcmc(Theta))
```

```
##
## Fraction in 1st window = 0.1
## Fraction in 2nd window = 0.5
##
## Alpha      mu      ss
## 1.7975 -0.8180 0.5608
```

```
median(mu)
```

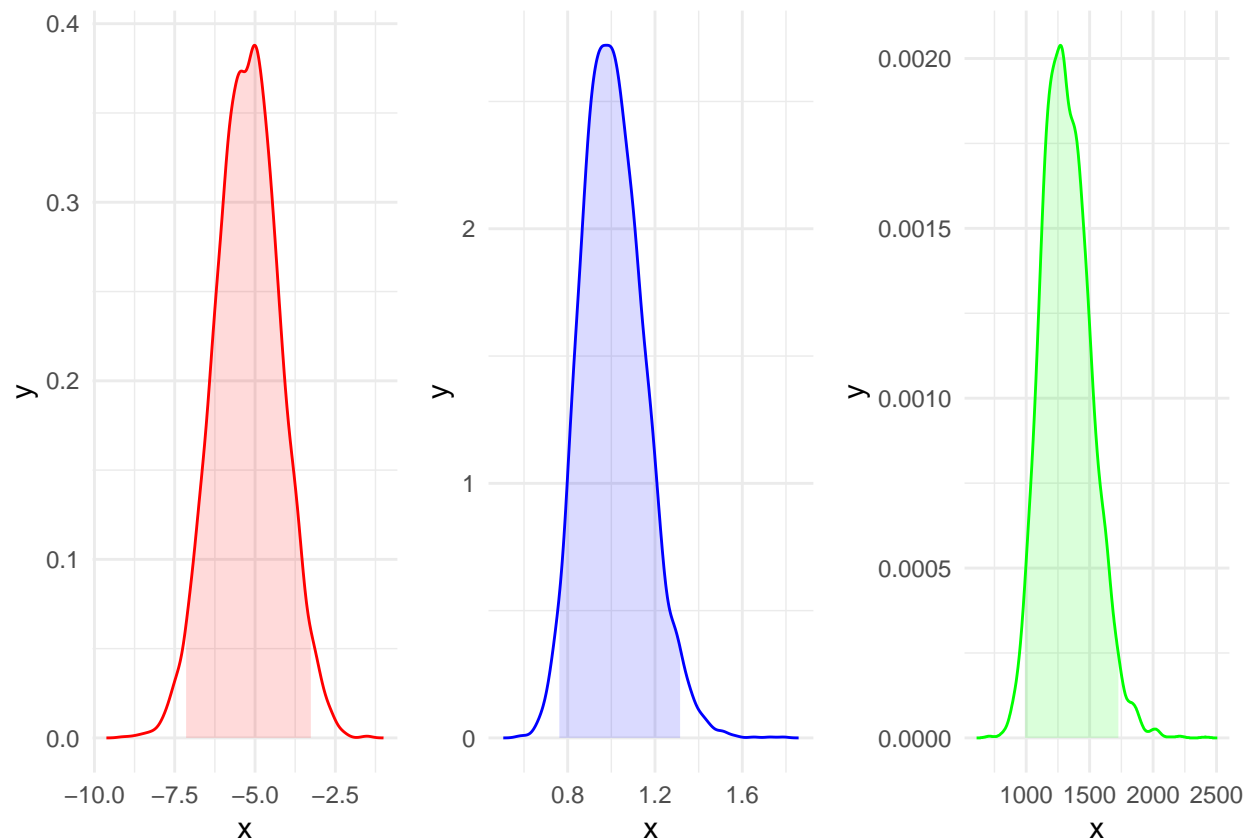
```
## [1] -5.218309
```

```
median(ss)
```

```
## [1] 1.000116
```

```
median(Alpha)
```

```
## [1] 1302.815
```

Normal Model for the Treatment Group

```
hers1 <- hers[ which(hers[,2] == 1), 1:2]

B <- 10000

ss <- Alpha <- mu <- rep(NA,B)
ss[1] <- Alpha[1] <- mu[1] <- 1
v <- 1
n <- nrow(hers1)

a <- b <- cc <- d <- .0001

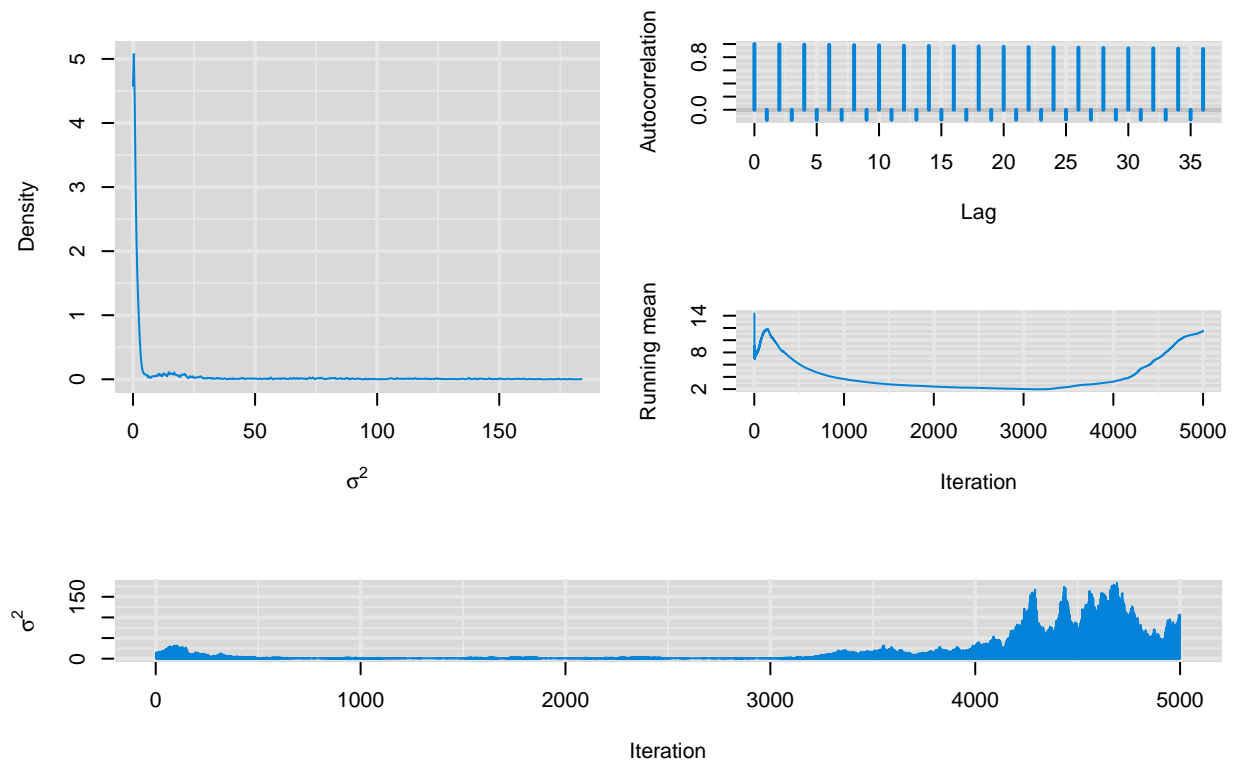
set.seed(32717)
for (t in 2:B) {
  Alpha[t] <- rinvgamma(1, (n)/2 + a, b + (1/2 * ss[t-1]) * sum( (hers1$y - mu[t-1])^2 ))
  mu[t] <- rnorm(1, mean(hers1$y), sqrt( Alpha[t-1]*ss[t-1] / n ))
  ss[t] <- rinvgamma( 1, (n/2)+ cc, d + (1/(2*Alpha[t])) * sum( (hers1$y - mu[t])^2 ) )
}

# Burn in
```

```
Alpha <- matrix( tail(Alpha,B/2), ncol = 1);colnames(Alpha) <- "alpha"
mu <- mu1a <- matrix( tail(mu,B/2), ncol = 1);colnames(mu) <- "mu"
ss <- matrix( tail(ss,B/2), ncol = 1);colnames(ss) <- "sigma^2"

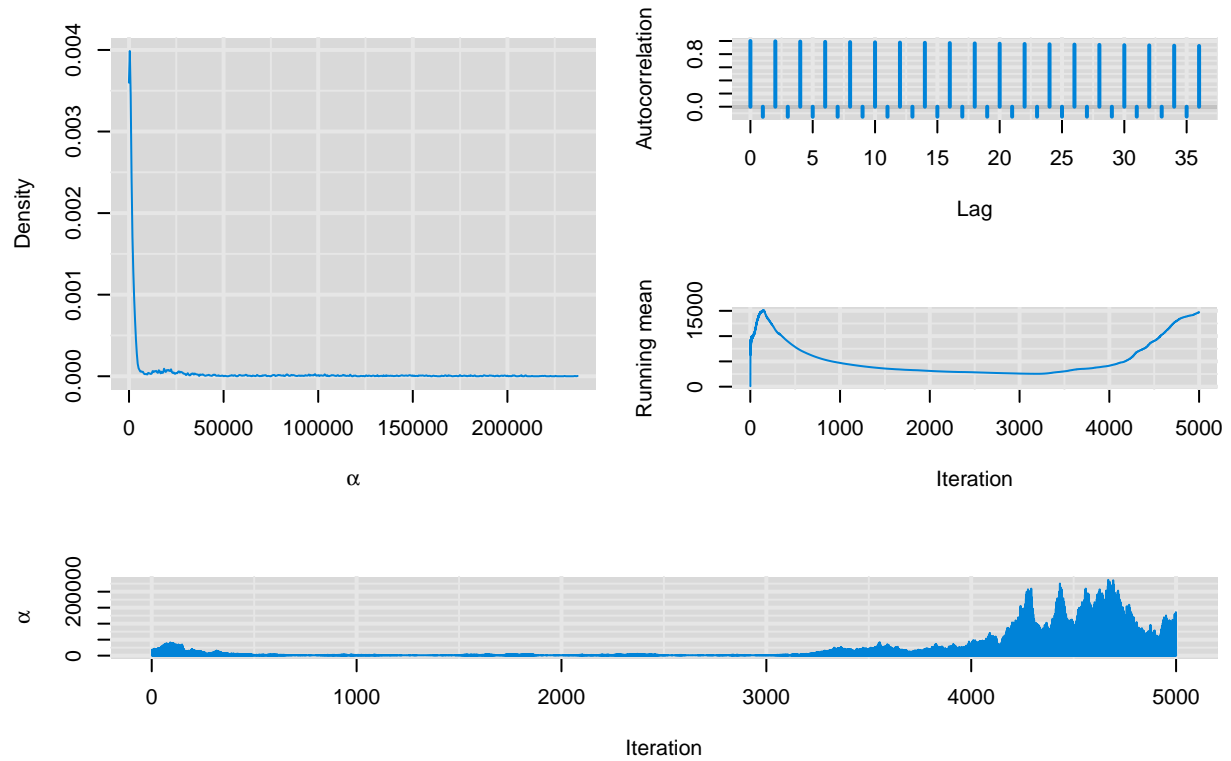
mcmcplot1( ss , greek = T )
```

Diagnostics for σ^2



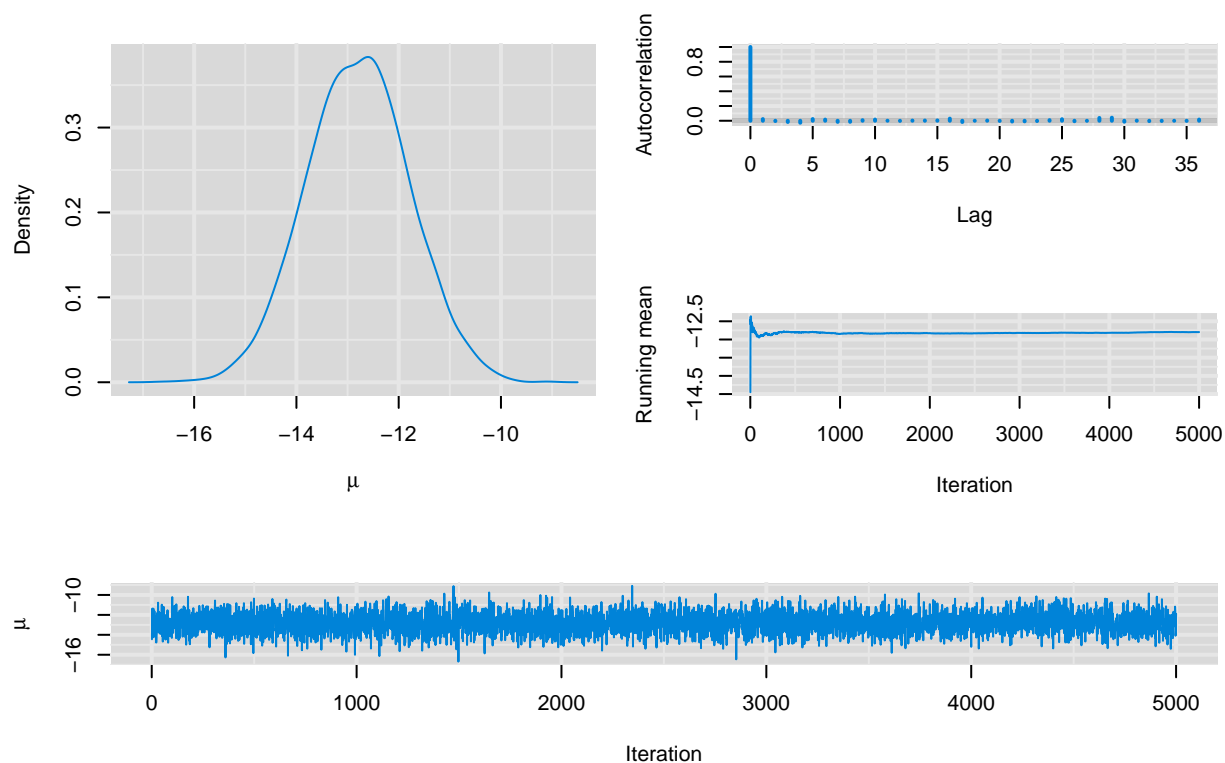
```
mcmcplot1( Alpha, greek = T )
```

Diagnostics for α



```
mcmcplot1( mu , greek = T)
```

Diagnostics for μ



```
Theta <- matrix( c(Alpha,mu,ss) , ncol = 3 ); colnames(Theta) <- c("Alpha","mu","ss")
t(apply(Theta, 2, quantile, probs = c(0.5, 0.025, 0.975)))
```

```
##           50%           2.5%           97.5%
## Alpha 1280.2611319  10.905214820 153103.55105
## mu    -12.7897880 -14.715393912  -10.82047
## ss      0.9992907  0.008424543   118.73576
```

```
geweke.diag(mcmc(Theta))
```

```
##
## Fraction in 1st window = 0.1
## Fraction in 2nd window = 0.5
##
## Alpha      mu      ss
## -1.2756 -0.8017 -1.2331
```

```
median(mu)
```

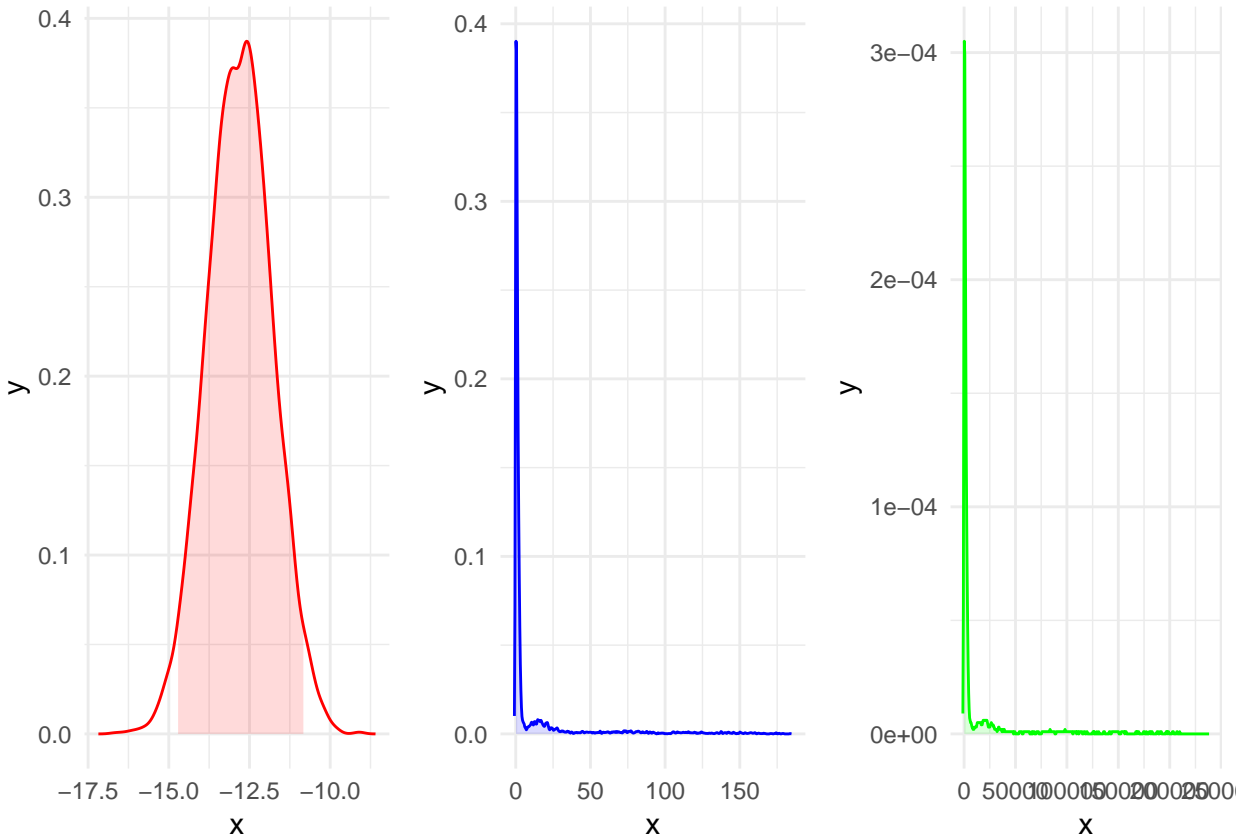
```
## [1] -12.78979
```

```
median(ss)
```

```
## [1] 0.9992907
```

```
median(Alpha)
```

```
## [1] 1280.261
```



σ^2 and α are not converging well, so we select some new values for a , b , c , d and reevaluate.

```
ss <- Alpha <- mu <- rep(NA,B)
ss[1] <- Alpha[1] <- mu[1] <- 1
v <- 1
n <- nrow(hers1)

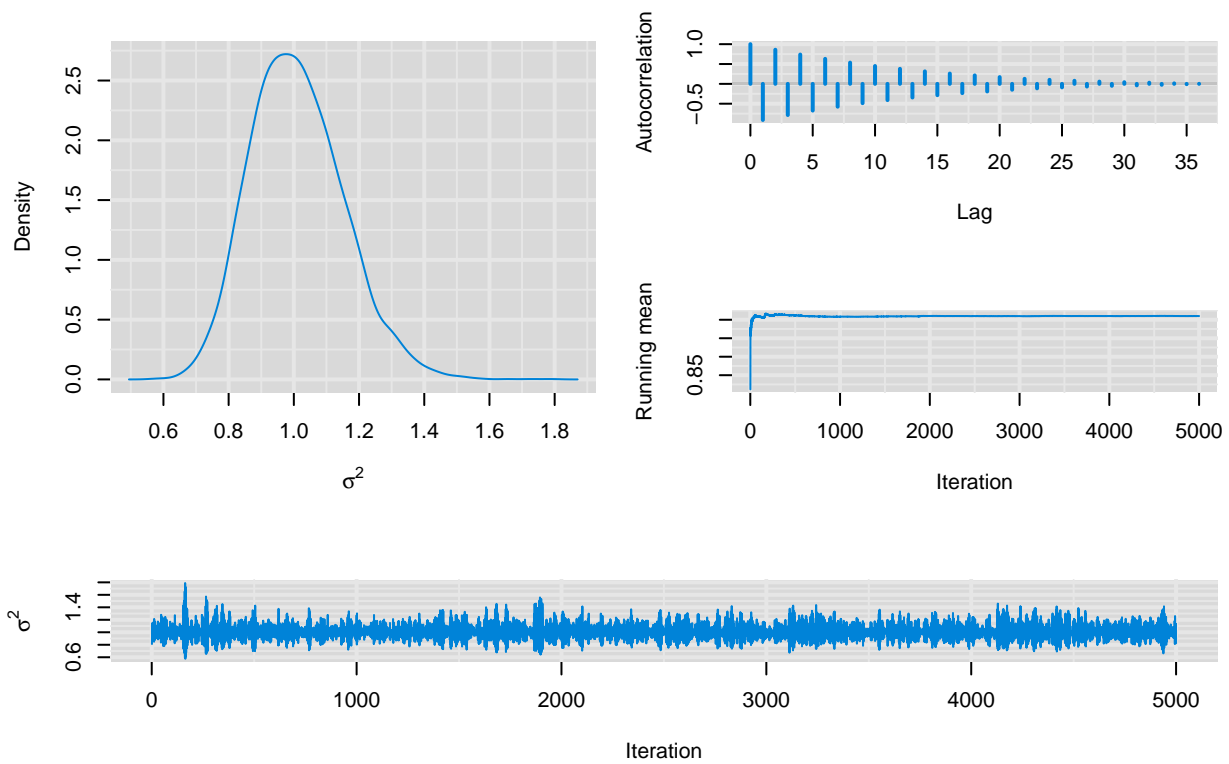
a <- b <- .1
cc <- d <- 50

set.seed(32717)
for (t in 2:B) {
  Alpha[t] <- rinvgamma(1, (n)/2 + a, b + (1/2 * ss[t-1]) * sum( (hers1$y - mu[t-1])^2 ))
  mu[t] <- rnorm(1, mean(hers1$y), sqrt(Alpha[t-1]*ss[t-1] / n))
  ss[t] <- rinvgamma(1, (n)/2 + cc, d + (1/(2*Alpha[t])) * sum( (hers1$y - mu[t])^2 ))
}

# Burn in
Alpha <- matrix( tail(Alpha,B/2), ncol = 1); colnames(Alpha) <- "alpha"
mu <- mu1b <- matrix( tail(mu,B/2), ncol = 1); colnames(mu) <- "mu"
ss <- matrix( tail(ss,B/2), ncol = 1); colnames(ss) <- "sigma^2"

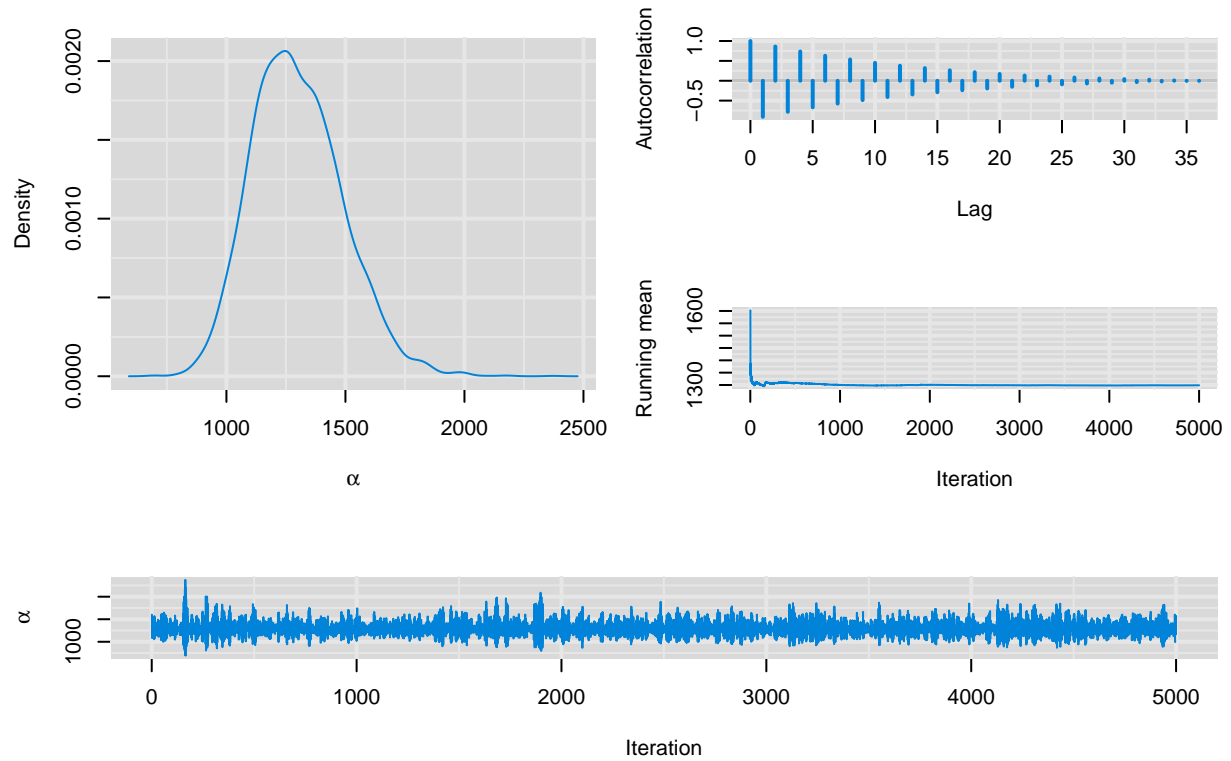
mcmcplot1( ss, greek = T )
```

Diagnostics for σ^2



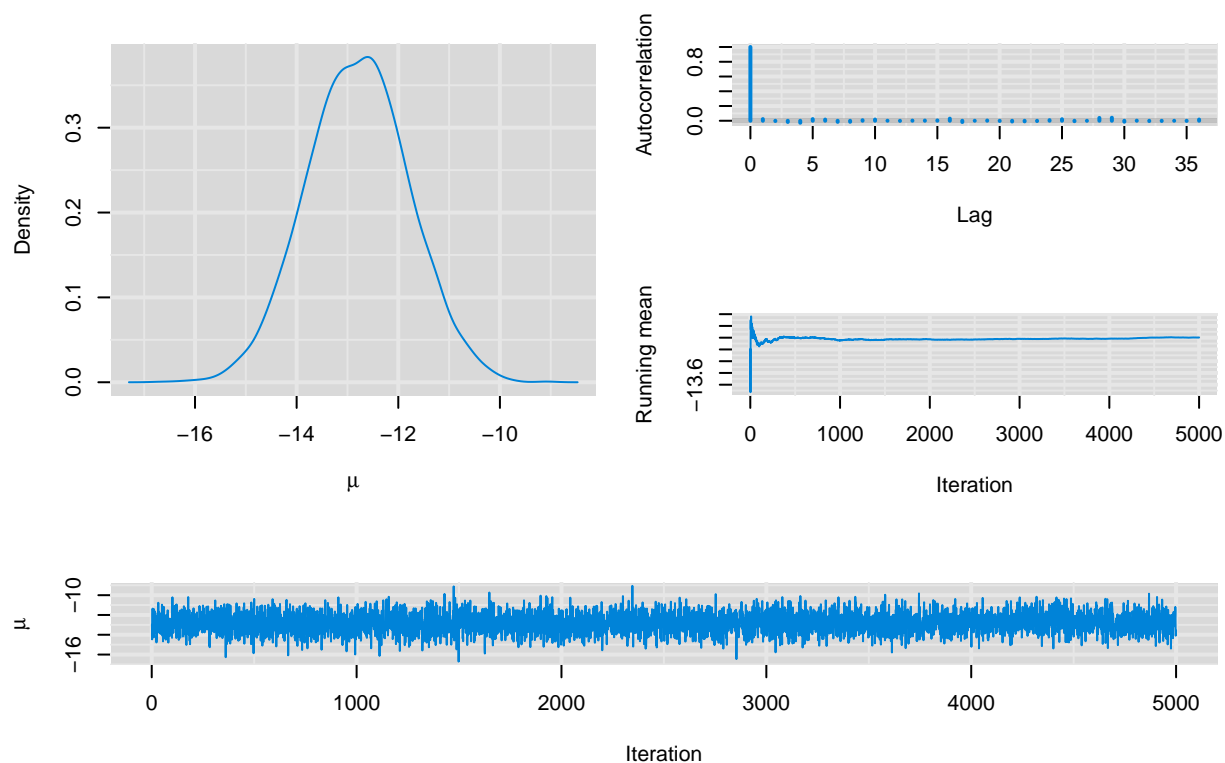
```
mcmcplot1( Alpha, greek = T )
```

Diagnostics for α



```
mcmcplot1( mu , greek = T)
```

Diagnostics for μ



```
Theta <- matrix( c(Alpha,mu,ss) , ncol = 3 ); colnames(Theta) <- c("Alpha","mu","ss")
t(apply(Theta, 2, quantile, probs = c(0.5, 0.025, 0.975)))
```

```
##           50%          2.5%          97.5%
## Alpha 1280.7226423  972.3334542 1701.312203
## mu    -12.7904048 -14.7209148 -10.814712
## ss      0.9999834  0.7595289  1.317094
```

```
geweke.diag(mcmc(Theta))
```

```
##
## Fraction in 1st window = 0.1
## Fraction in 2nd window = 0.5
##
## Alpha      mu      ss
## 1.7968 -0.8180 0.5497
```

```
median(mu)
```

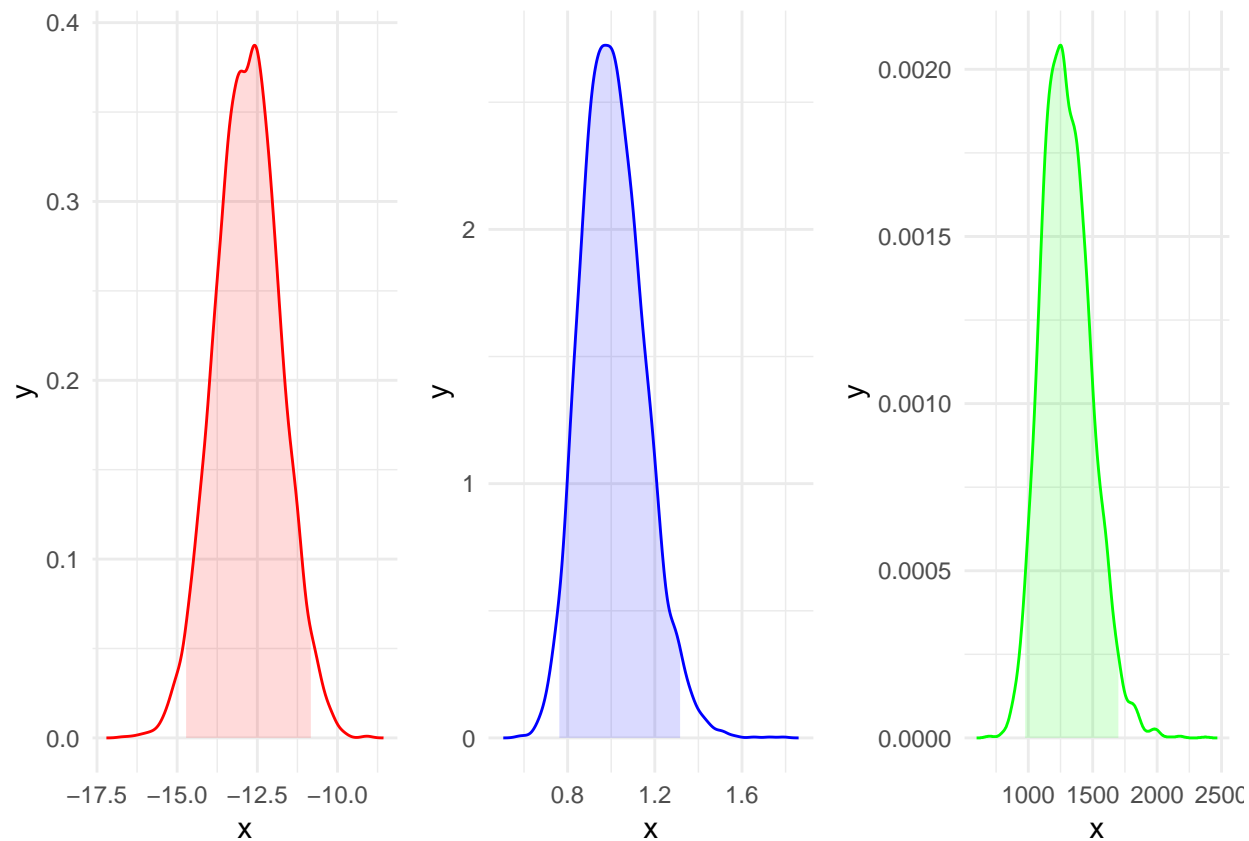
```
## [1] -12.7904
```

```
median(ss)
```

```
## [1] 0.9999834
```

```
median(Alpha)
```

```
## [1] 1280.723
```

Difference in means

```
quantile(mu0a - mu1a, probs = c(0.5, 0.025, 0.975))
```

```
##      50%      2.5%      97.5%
## 7.559315 4.811147 10.335666
```

```
quantile(mu0b - mu1b, probs = c(0.5, 0.025, 0.975))
```

```
##      50%      2.5%      97.5%
## 7.572095 7.569030 7.575144
```

Zero is not in either credible interval so we conclude they come from different distributions.

$$\mathcal{L}(y|\mu, \sigma^2) \propto \prod_{i=1}^n \frac{\Gamma\left(\frac{n+2}{2}\right)}{\Gamma\left(\frac{n+1}{2}\right) \sqrt{(n-1)\sigma^2}} \left[1 + \frac{(y_i - \mu)^2}{(n-1)\sigma^2}\right]^{-\frac{n}{2}}$$

$$p(\mu, \sigma^2, y) \propto \exp\left[-\frac{n}{2} \sum_{i=1}^n \log\left(1 + \frac{(y_i - \mu)^2}{(n-1)\sigma^2}\right)\right]$$

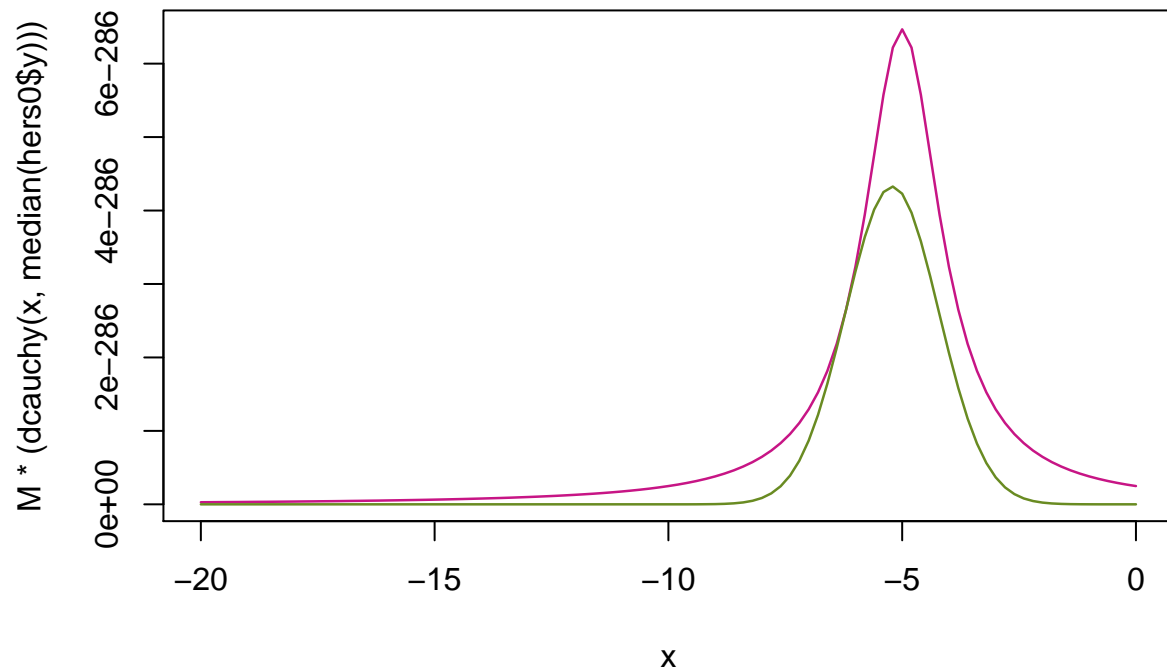
```
fy <- function( y , mu ){
  n <- length(y)
  s <- sd(y)
  return( sapply( mu , function(MU) exp( (-n/2) *
    sum( log( 1 + (y-MU)^2/((n-1)*(s^2)) )) )) ) ) }

fy2 <- function( x , y ){ fy(hers0$y, x) / dcauchy(x, median(hers0$y) ) }
optimize( fy2 , lower = -40, upper = 10, maximum=T )

## $maximum
## [1] -6.191352
##
## $objective
## [1] 2.03157e-285

M <- optimize( fy2 , lower = -40, upper = 10, maximum=T )$objective

curve( M * ( dcauchy(x, median(hers0$y) ) ) , col = "mediumvioletred",lwd = 1.25 , from = -20, to =
curve( fy(hers0$y, x) , from = -20, to = 0 , add=T, col = "olivedrab4" ,lwd = 1.25)
```



```

Mu <- rep(NA, 10000)
t <- 1
count <- 1

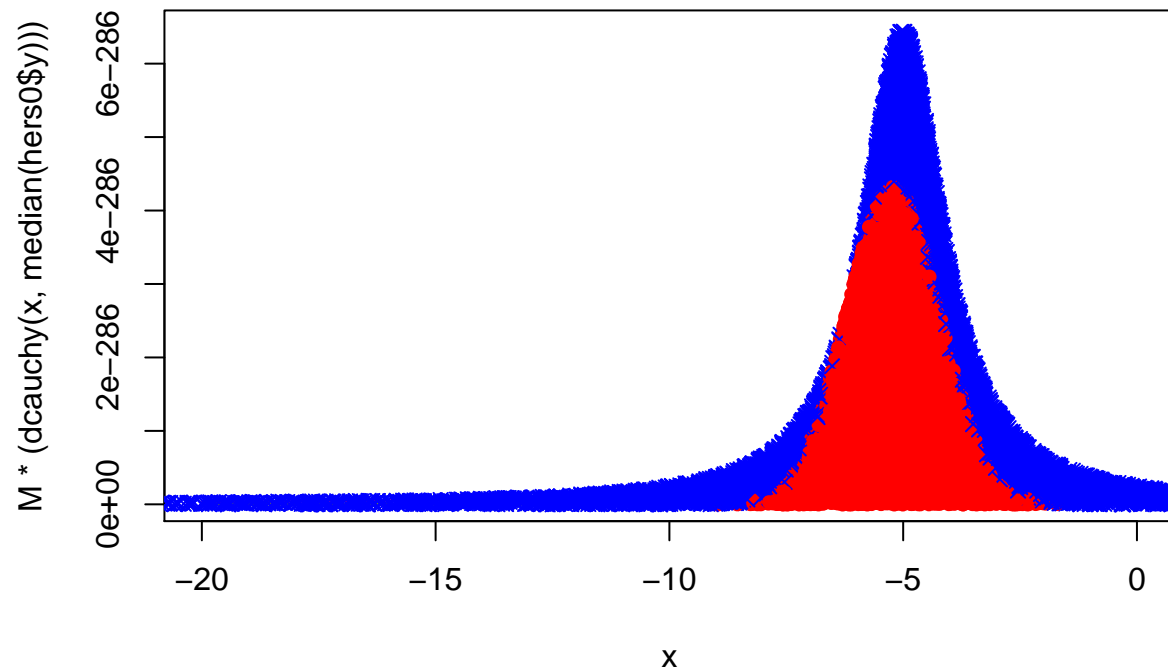
curve( M * ( dcauchy(x, median(hers0$y) ) ) ,      col = "mediumvioletred",lwd = 1.25 , from = -20, to =
curve( fy(hers0$y, x) , from = -20, to = 0 , add=T,      col = "olivedrab4" ,lwd = 1.25)

set.seed(50814)
while(t < 10001){

  tb <- rcauchy(1, median(hers0$y) )
  U <- runif(1)
  r <- fy(hers0$y,tb) / (M*dcauchy(tb, median(hers0$y) ))

  if(U < r){
    Mu[t] <- tb
    t <- t + 1
  }
  points( tb , M*U*dcauchy(tb, median(hers0$y) ) , col = "red" , pch = 16) } else {
  points( (tb) , M*U*dcauchy(tb, median(hers0$y) ) , col = "blue" , pch = 4) }
  count <- count + 1 }

```



```
tail(Mu)
```

```
## [1] -6.289775 -5.016245 -6.400210 -4.569171 -7.864230 -6.727801
```

```
median(Mu)
```

```
## [1] -5.203712
```

```
Mu <- Mu0 <- matrix( Mu, ncol = 1);colnames(Mu) <- "mu"
geweke.diag(mcmc(Mu))
```

```
##
```

```
## Fraction in 1st window = 0.1
```

```
## Fraction in 2nd window = 0.5
```

```
##
```

```
##      mu
```

```
## 0.09529
```

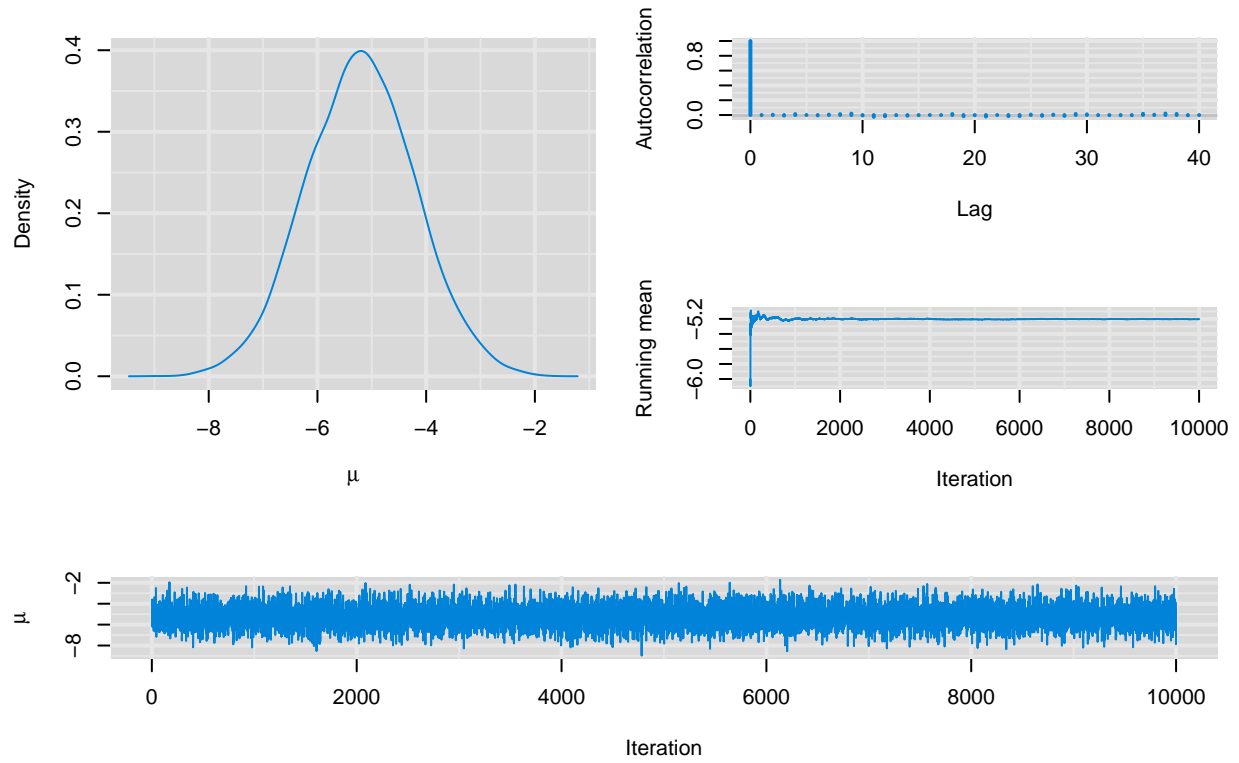
```
t(apply(Mu, 2, quantile, probs = c(0.5, 0.025, 0.975)))
```

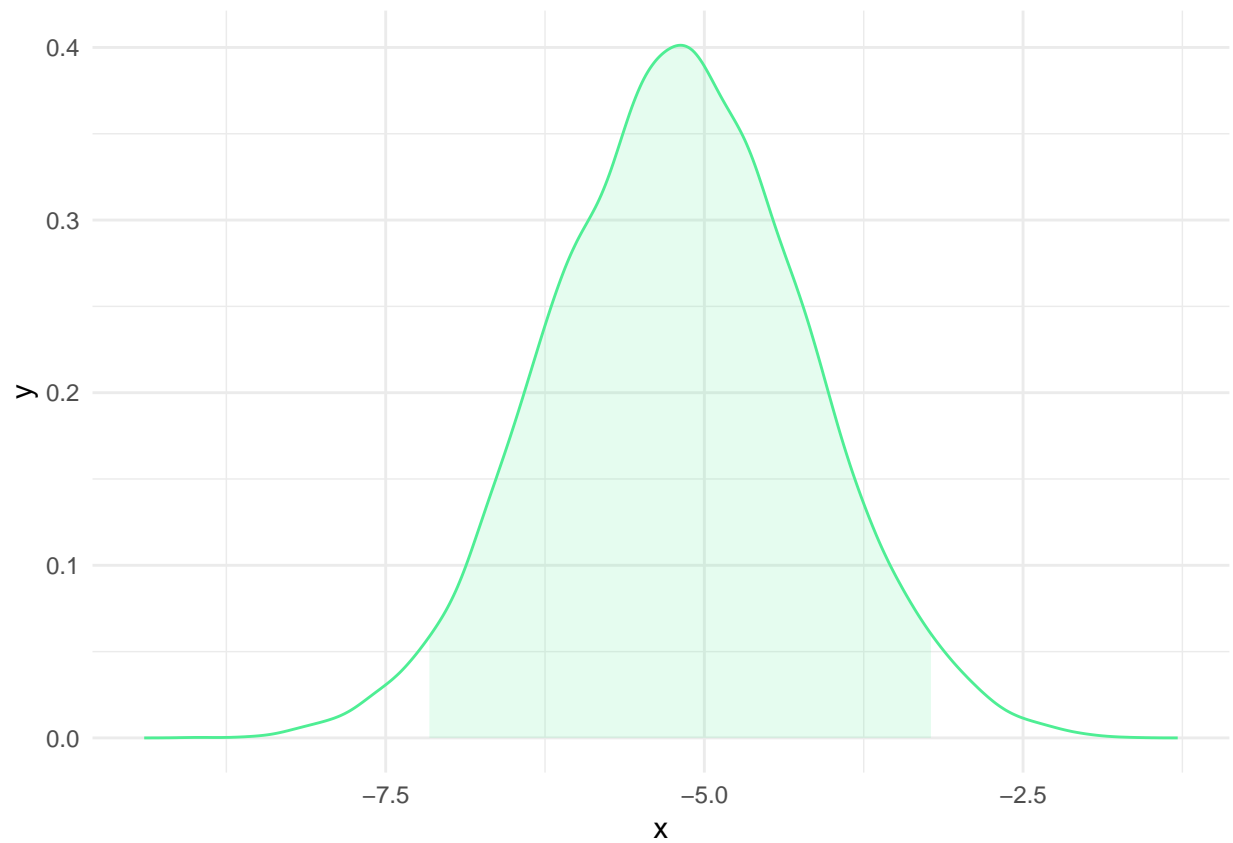
```
##      50%      2.5%      97.5%
```

```
## mu -5.203712 -7.164455 -3.219134
```

```
mcmcplot1( Mu , greek = T )
```

Diagnostics for μ

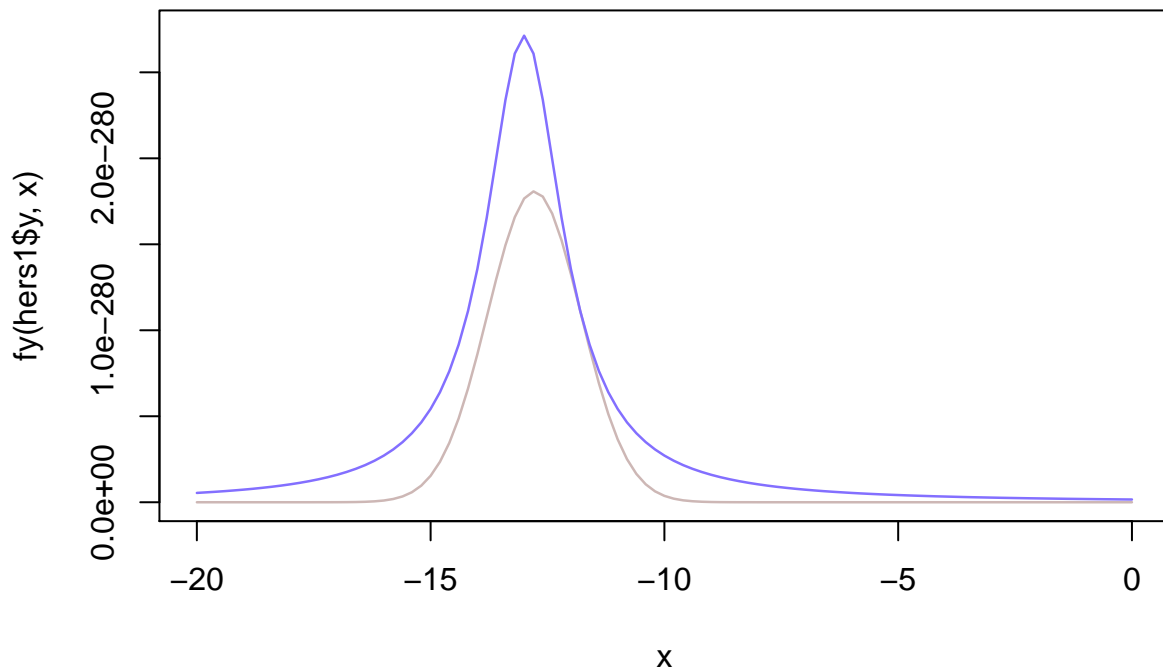




```
fy2 <- function( x , y ){ fy(hers1$y, x) / dcauchy(x, median(hers1$y) ) }
optimize( fy2 , lower = -40, upper = 10, maximum=T )
```

```
## $maximum
## [1] -11.80364
##
## $objective
## [1] 8.524237e-280
```

```
M <- optimize( fy2 , lower = -40, upper = 10, maximum=T )$objective
curve( fy(hers1$y, x) , from = -20, to = 0 , ylim = c(0,2.75e-280), col = "mistyrose3" ,lwd = 1.25)
curve( M*( dcauchy(x, median(hers1$y) ) ) ,add = T , col = "slateblue1",lwd = 1.25)
```



```

Mu <- rep(NA, 10000)
t <- 1
count <- 1

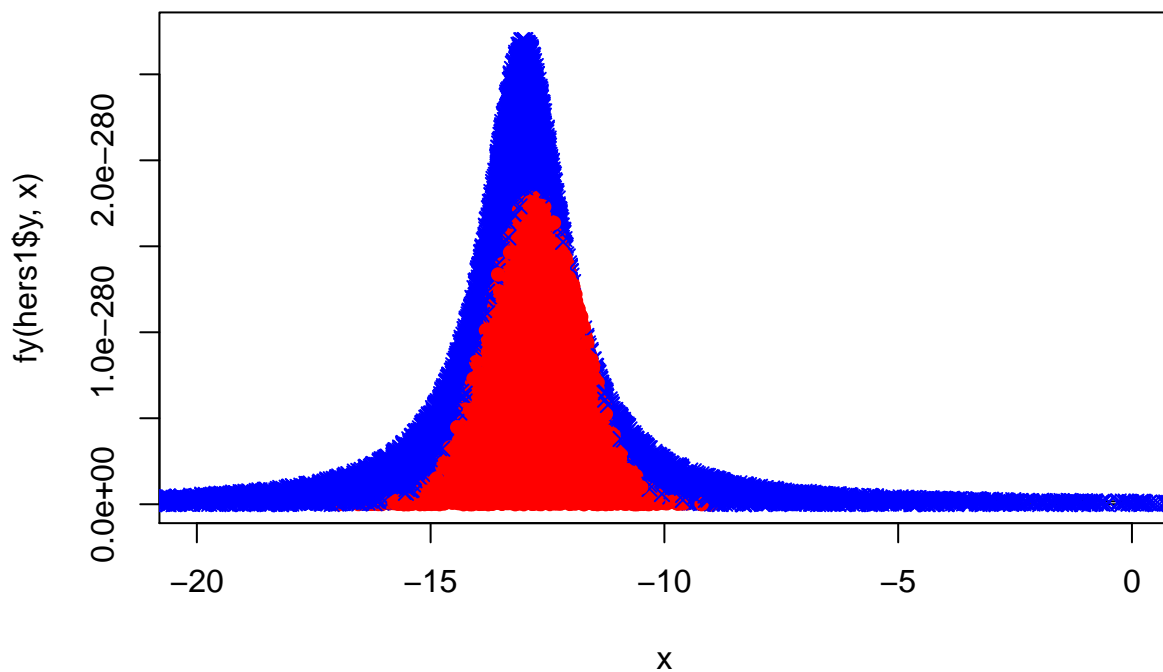
curve( fy(hers1$y, x) , from = -20, to = 0 , ylim = c(0,2.75e-280))
curve( M*( dcauchy(x, median(hers1$y) ) ) ,add = T)

set.seed(50814)
while(t < 10001){

  tb <- rcauchy(1, median(hers1$y) )
  U <- runif(1)
  r <- fy(hers1$y,tb) / (M*dcauchy(tb, median(hers1$y) ))

  if(U < r){
    Mu[t] <- tb
    t <- t + 1
  }
  points( tb , M*U*dcauchy(tb, median(hers1$y) ), col = "red" , pch = 16) } else {
  points( (tb) , M*U*dcauchy(tb, median(hers1$y) ), col = "blue" , pch = 4) }
  count <- count + 1 }

```



```
median(Mu)
```

```
## [1] -12.77387
```

```
Mu <- Mu1 <- matrix( Mu, ncol = 1); colnames(Mu) <- "mu"
geweke.diag(mcmc(Mu))
```

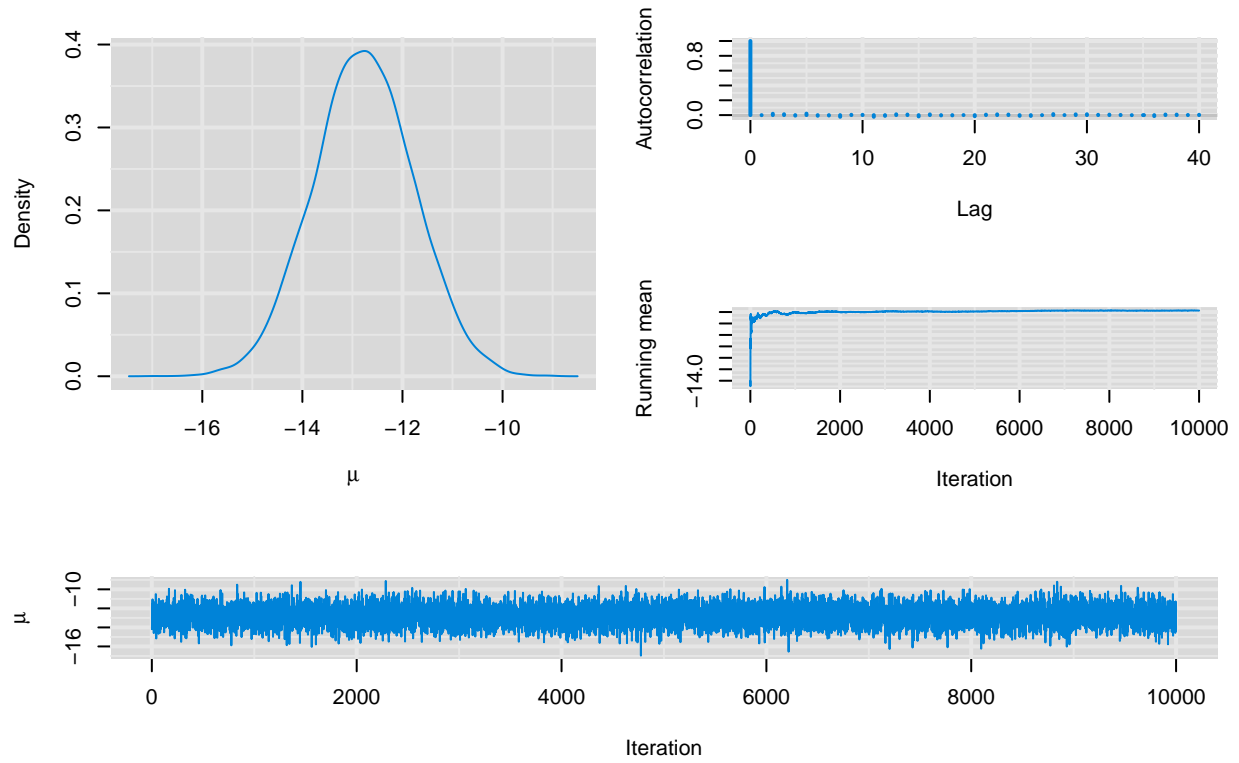
```
##
## Fraction in 1st window = 0.1
## Fraction in 2nd window = 0.5
##
##      mu
## -1.361
```

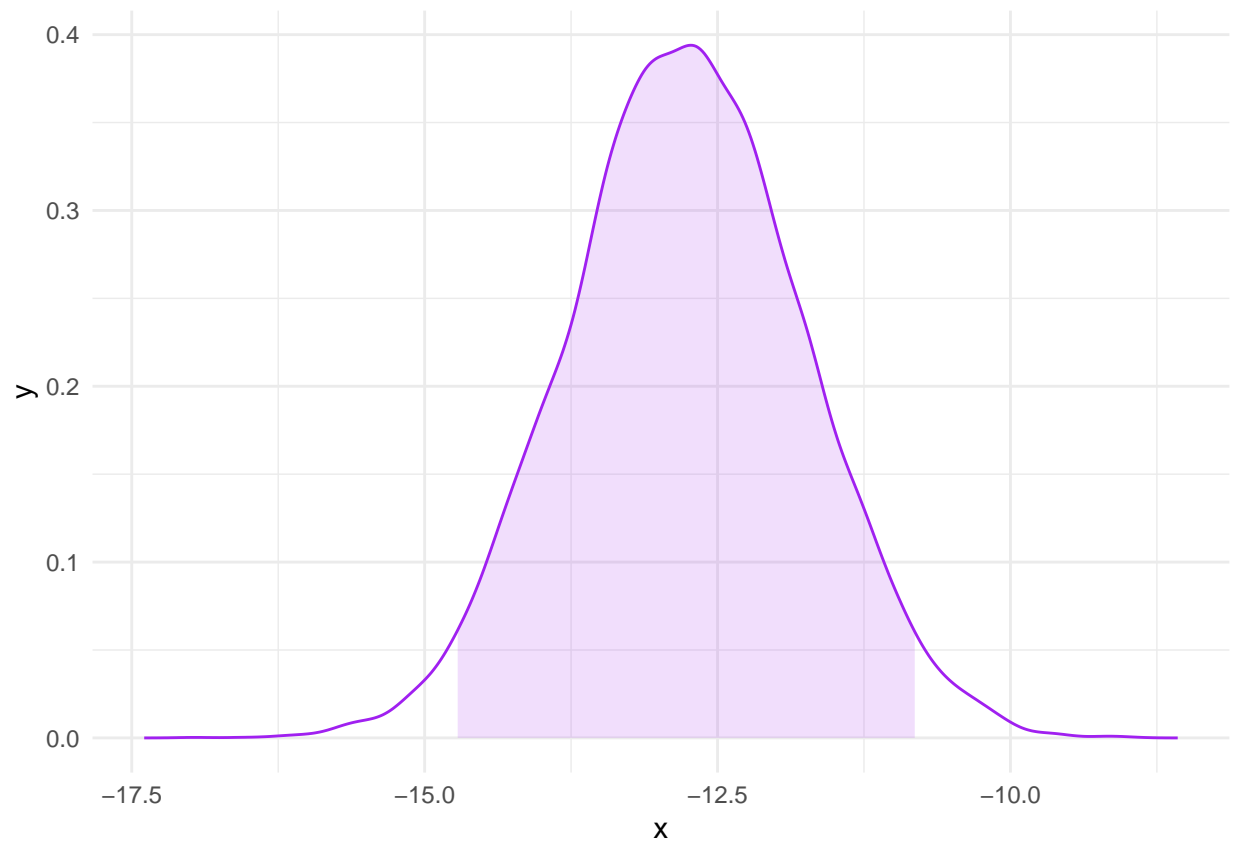
```
t(apply(Mu, 2, quantile, probs = c(0.5, 0.025, 0.975)))
```

```
##           50%       2.5%      97.5%
## mu -12.77387 -14.72148 -10.81254
```

```
mcmcplot1( Mu , greek = T )
```


Diagnostics for μ





For both t -likelihood models based on the visual inspection the parameter appears to converge well. It is backed up by the Geweke diagnostic; with both scores with an absolute value less than two.

```
quantile(Mu0 - Mu1, probs = c(0.5, 0.025, 0.975))
```

```
##      50%      2.5%      97.5%
##  7.592700  4.806914 10.358324
```

Zero is not in either credible interval so we conclude they come from different distributions.

The credible intervals for each respective group and for each model, normal or t , appear to be incredibly similar.