

Metropolis Hasting Algorithm

Generate samples from distribution $x^5 e^{-x}$ by using proposal distribution as $\log\{\text{normal LN}(X_t;1)\}$

We took log normal as our proposal distribution to generate samples for our desired distribution. From the markov chain, the burn in period seems to be first 20 variables which is represented by red line, which represent the non equilibrium state. Post that the markov chain seems to be reaching in equilibrium, moving in the range from 3 to 4.

```
target_dist <- function(x){
  if(x<0){
    return(0)}
  else {
    return((x^5)*exp(-x))
  }
}

#Proposal Distribution
proposed_dist <- function(rv){
  rlnorm(1, meanlog = log(rv), sdlog = 1) #log normal
}

metropolis_algorithm <- function(samp,rand_no) {
  curr_x <- rand_no #Current random value
  post_val <- rep(0,times = samp) #Number of Posterior value which is accepted.
  prop_val <- c() #Generate values from the proposal distribution.
  cnt <- 1

  #Iterate or generate 2000 points for target distribution using proposal distribution
  while(cnt <= samp){

    #Generate a new value from proposal distribution based on current value and sd 1.
    #Basically move to the near point. it is kind of markov move which is dependent on
    #current value.

    #Generate next RV based on the current RV.
    prop_x <- proposed_dist(curr_x)
    U <- runif(1, 0, 1)

    #posterior <- likelyhood* prior
    posterior_prop <- target_dist(prop_x)*dlnorm(curr_x,mean = log(prop_x),sd = 1)
    posterior_curr <- target_dist(curr_x)*dlnorm(prop_x,mean = log(curr_x),sd = 1)

    ifelse(U < min(1,(posterior_prop/posterior_curr)),post_val[cnt] <- prop_x,
           post_val[cnt] <- curr_x)

    curr_x <- post_val[cnt]
    prop_val[cnt] <- prop_x

    cnt <- cnt+1
  }
  return(list(post_val,prop_val))
}
```

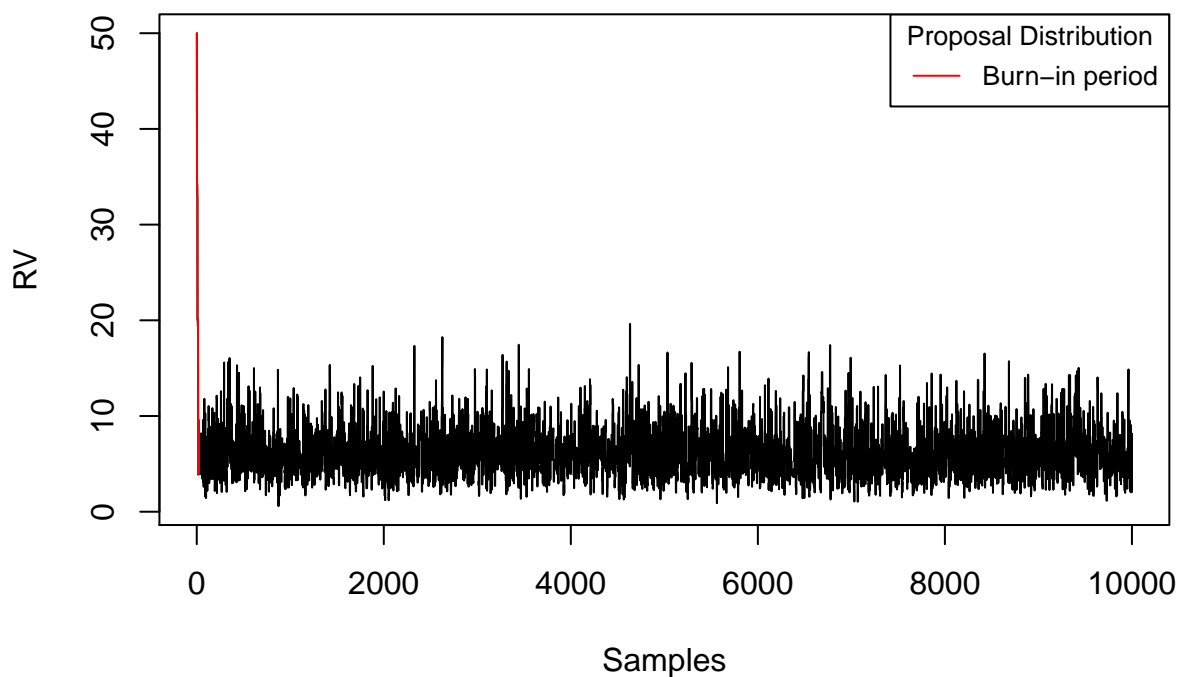
```

}

val1 <- metropolis_algorithm(10000,50)
{plot(1:10000,y=val1[[1]],type="l",xlab="Samples",ylab="RV",col="black")
  lines(x=1:20,y=val1[[1]][1:20],col="red")
title(main = "Markov Chain for generated samples using Log Normal\n as proposal")
legend("topright",legend=c("Burn-in period"),
col=c("red"), lty=1:2, cex=0.8,title="Proposal Distribution")
}

```

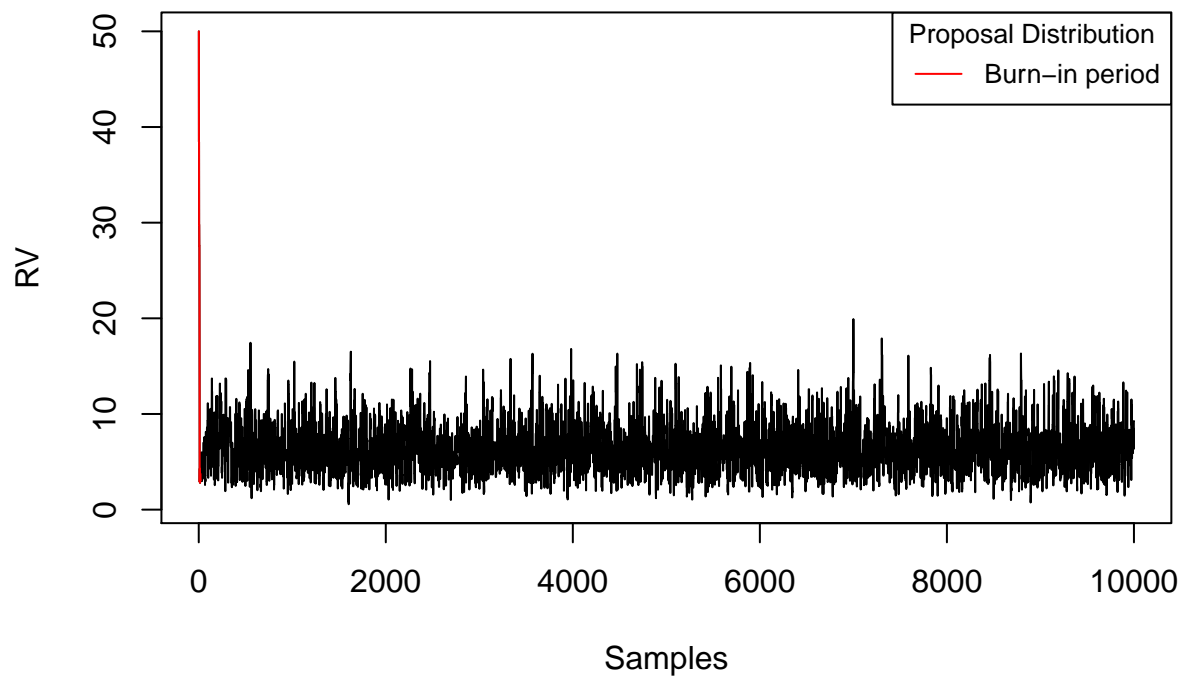
Markov Chain for generated samples using Log Normal as proposal



Chi Square distribution as proposal distribution

We took Chi Square as our proposal distribution to generate samples for our desired distribution. From the markov chain, the burn in period seems to be first 15 variables which is represented by red line, which represent the non equilibrium state. Post that the markov chain seems to be reaching in equilibrium, moving in the range from 4 to 6.

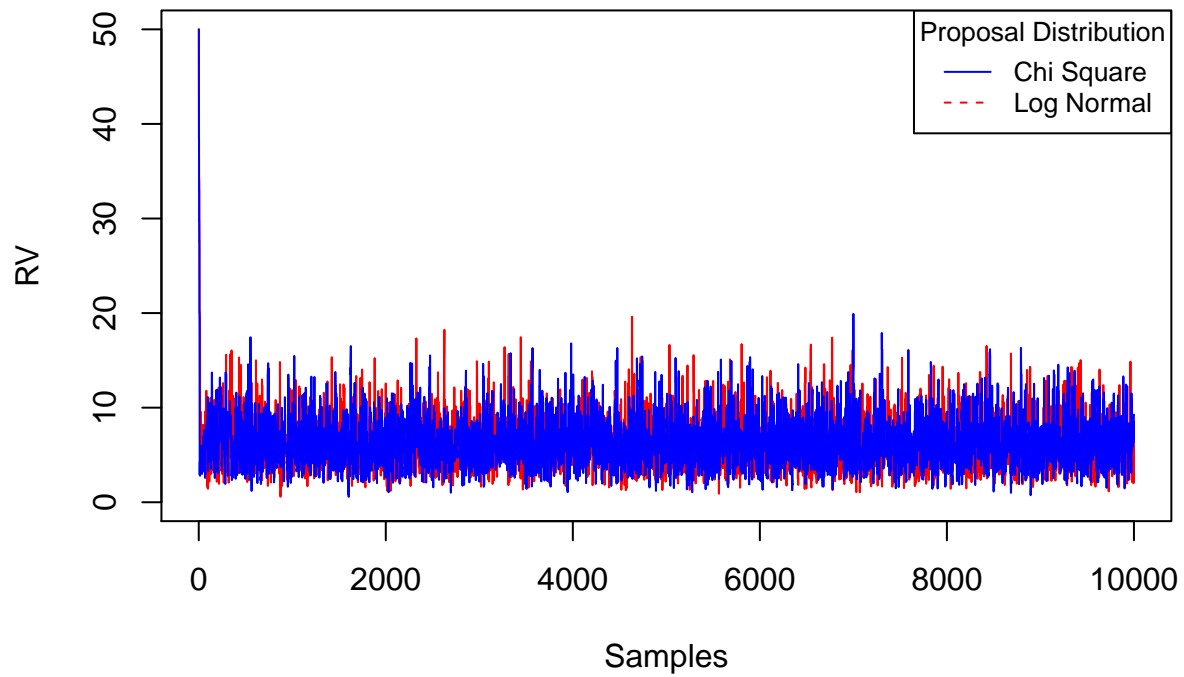
Markov Chain for generated samples using Chi Square as proposal



Comparison

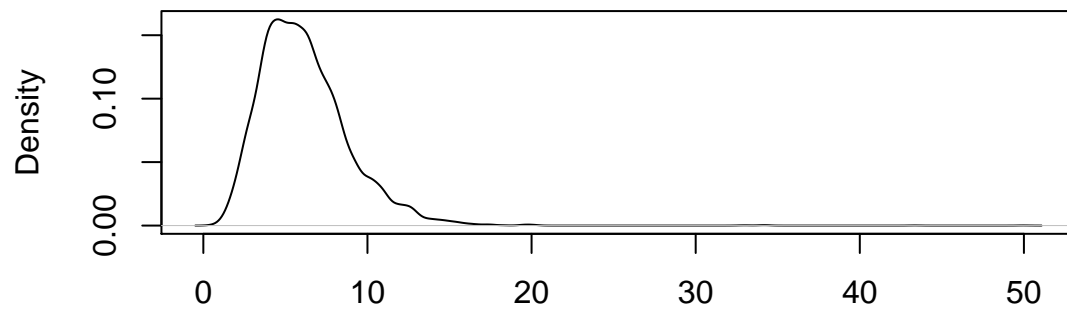
From the comparison of 2 markov chain, the samples generated by Chi Square seems to be converging better than Log normal.

Markov Chain for generated samples by Chi Square vs Log Normal

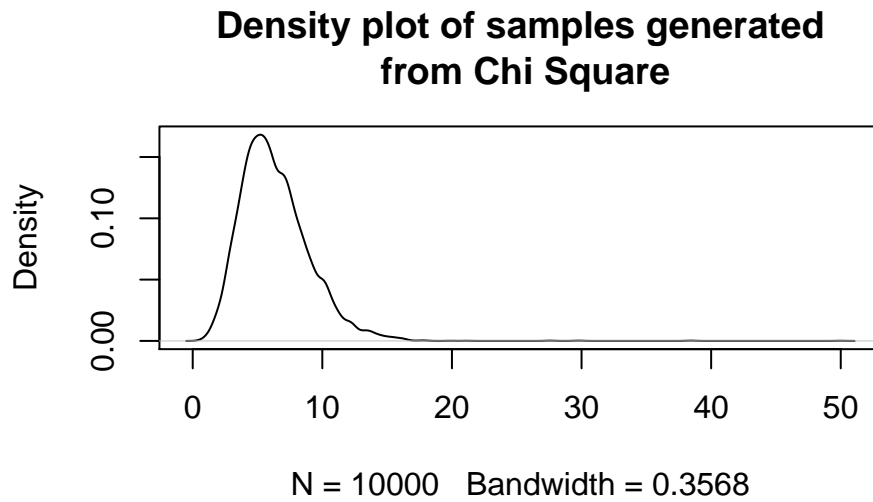


Below is the two density plot samples generated by log normal and chisquare

Density plot of samples generated from Log normal



N = 10000 Bandwidth = 0.3586



From the above two plot, it seems the chi square is more accurately representing our function than the log normal.

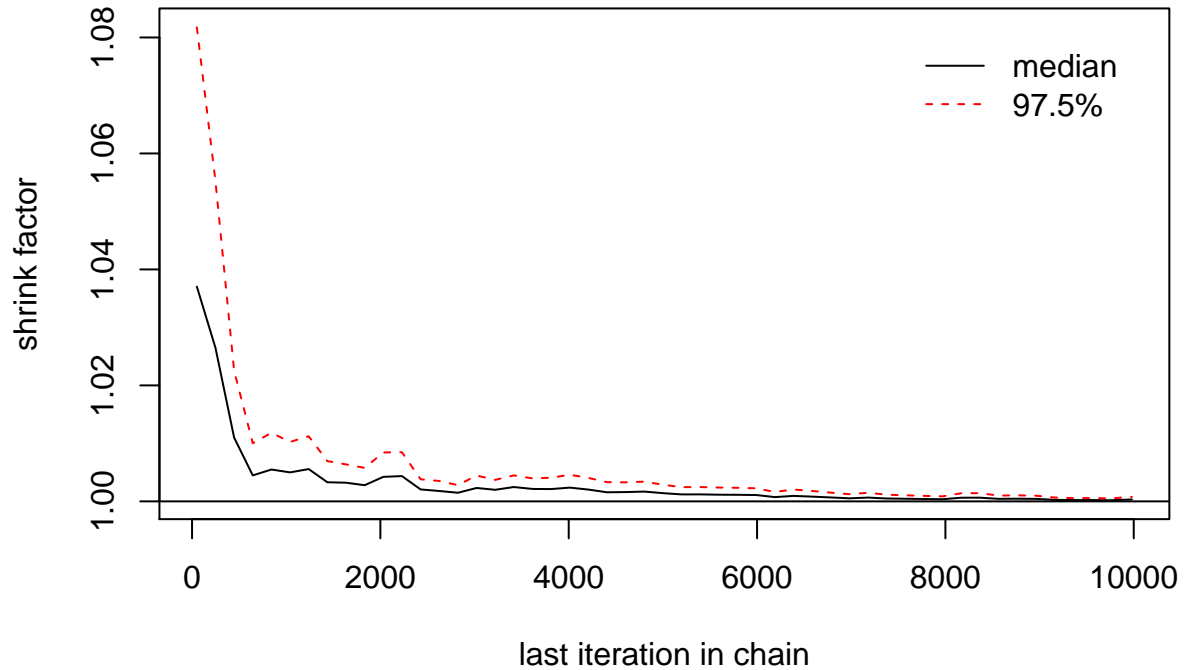
Generate 10 MCMC sequences using Chi Square as proposal function and use Gelman-Rubin method to analyze convergence.

Using Gelman-Rubin as convergence method , we get 'potential scale reduction factor'. Approximate convergence is diagnosed when the upper limit is close to 1. In our case it is 1. Hence we can assume that about markov chain does converges.

Potential scale reduction factors:

	Point est.	Upper C.I.
[1,]	1	1

Convergence of markov chain



Estimate $\int_0^\infty x.f(x)$

We can estimate the mean as:

$$E[x] = \int x f(x) dx \approx \frac{1}{j-i} \sum_{t=i+1}^j x_t$$

where j-i is the total number of iterations. Basically we are calculating mean.

Mean of samples by Log normal: 6.16981

Mean of samples by Chi Sqaure: 6.36461

On manually doing the integral of $\int_0^\infty x.f(x)$, we the the value as 6. The value of mean by log normal and chisquare is approximately closer to the generated value.

Gamma Distribution

The Gamma distribution equation is :

$$\Gamma(\alpha) = \int_0^\infty x^{\alpha-1} * e^{-x} dx$$

Comparing with our equation $f(x) = \int_0^\infty x^5 * e^{-x}$, we can deduce that $\alpha = 6$. The mean of the equation is α which is 6.