

# Metropolis-Hastings

*Seung Ah Ha, Jaymo Kim, Wonbin Song*

## Metropolis-Hastings

```
a <- 6
b <- 4

likelihood <- function(param){
  singlelikelihoods = dbeta(param, a, b, log = T)
}

prior <- function(param){
  pr = dunif(param, min=0, max=1, log = T)
  return(pr)
}

posterior <- function(param){
  return (exp(likelihood(param) + prior(param)))
}

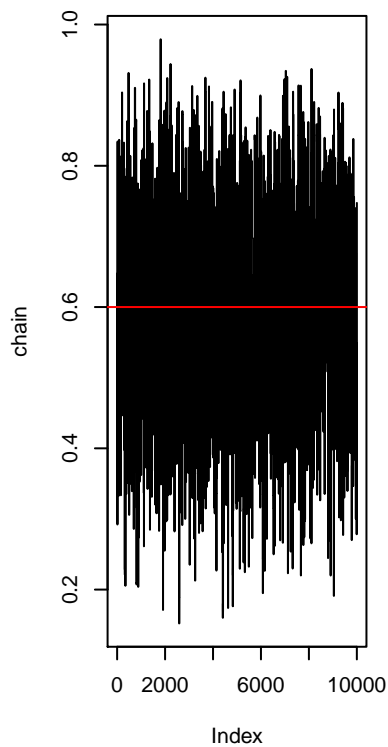
metropolis_MCMC <- function(startvalue, iterations, c){
  chain = array(dim = c(iterations+1,1))
  chain[1,] = startvalue
  for (i in 1:iterations){
    proposal = rbeta(1, c*chain[i,], c*(1-chain[i,]))
    probab = (dbeta(proposal,a,b)*dbeta(chain[i,],c*(1-proposal)))/
      (dbeta(chain[i,],a,b)*dbeta(proposal, c*chain[i,], c*(1-chain[i,])))
    if (runif(1) < probab){
      chain[i+1,] = proposal
    }else{
      chain[i+1,] = chain[i,]
    }
  }
  return(chain)
}

startvalue <- runif(1, 0, 1)
chain=metropolis_MCMC(startvalue, 10000, c=1)

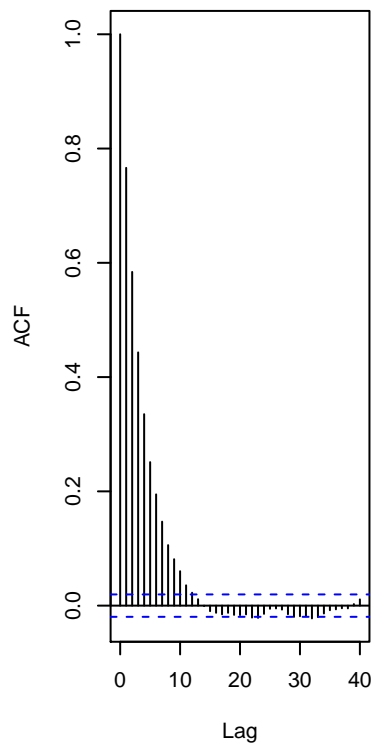
test <- rbeta(10000, 6, 4)

# Performance of the sampler: c = 1
par(mfrow=c(1,3))
plot(chain, type='l', main = "Trace plot of Chain, c = 1")
abline(h=0.6, col="red")
acf(chain, main = "Autocorrelation plot of Chain, c = 1")
hist(chain, xlab = "Samples from chain", main = "Histogram of Chain, c = 1")
```

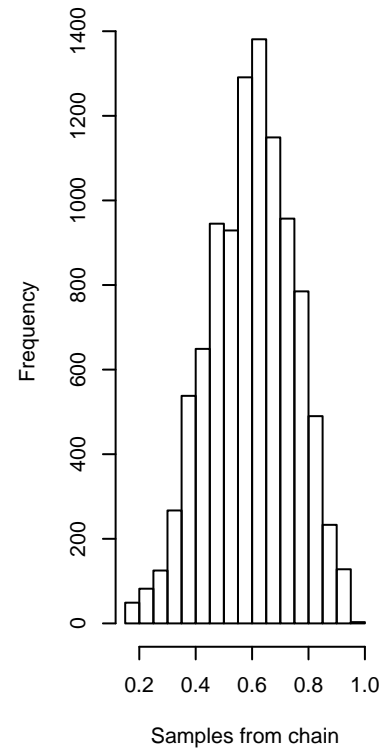
Trace plot of Chain, c = 1



Autocorrelation plot of Chain, c

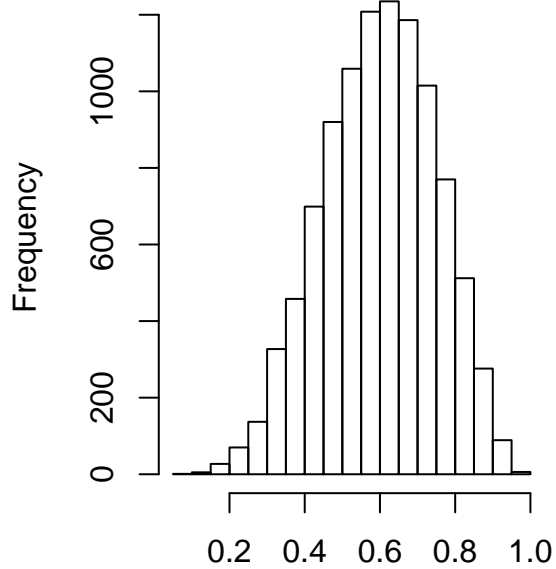


Histogram of Chain, c = 1



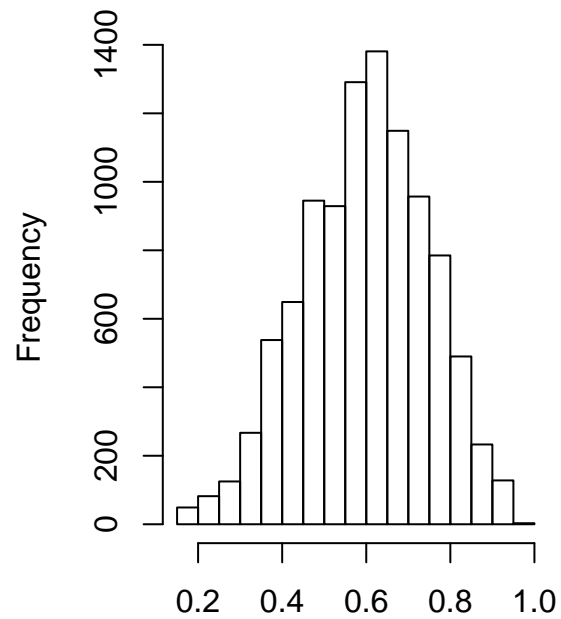
```
# Histogram of the chain vs. the target distribution Beta(6,4)
par(mfrow=c(1,2))
hist(test, xlab = "Samples from Beta(6, 4)", main = "Histogram of Beta(6, 4)")
hist(chain, xlab = "Samples from Chain", main = "Histogram of Chain, c = 1")
```

### Histogram of Beta(6, 4)



Samples from Beta(6, 4)

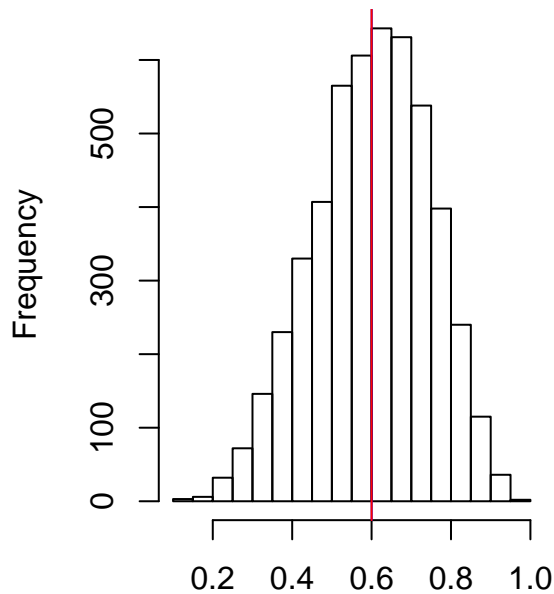
### Histogram of Chain, c = 1



Samples from Chain

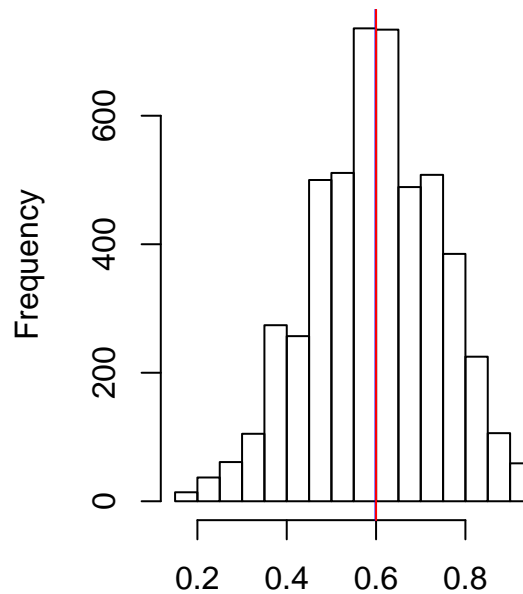
```
# Histogram of the target distribution Beta(6,4) vs. the Chain after burnIn
# Putting burnIn in
burnIn = 5000
test2 <- rbeta(5000, 6, 4) # true mean: 6/10 = 0.6
acceptance1 = 1-mean(duplicated(chain[-(1:burnIn),]))
hist(test2, xlab = "Samples from Beta(6, 4)", main = "Histogram of Beta(6, 4)")
abline(v=mean(test2), col="blue")
abline(v=0.6, col="red")
hist(chain[-(1:burnIn),1], xlab = "Samples from chain", main = "Histogram of Chain, c = 1")
abline(v = mean(chain[-(1:burnIn),1]), col="blue")
abline(v=0.6, col="red")
```

### Histogram of Beta(6, 4)



Samples from Beta(6, 4)

### Histogram of Chain, c = 1



Samples from chain

```
# Kolmogorov-Smirnov statistic
```

```
ks.test(chain[-(1:burnIn),1], test2)
```

```
## Warning in ks.test(chain[-(1:burnIn), 1], test2): p-value will be
## approximate in the presence of ties
```

```
##
```

```
## Two-sample Kolmogorov-Smirnov test
```

```
##
```

```
## data: chain[-(1:burnIn), 1] and test2
```

```
## D = 0.048473, p-value = 1.579e-05
```

```
## alternative hypothesis: two-sided
```

```
# c = 0.1 / c = 2.5 / c = 10
```

```
chain1=metropolis_MCMC(startvalue, 10000, c=0.1)
```

```
chain2=metropolis_MCMC(startvalue, 10000, c=2.5)
```

```
chain3=metropolis_MCMC(startvalue, 10000, c=10)
```

```
acceptance1 = 1-mean(duplicated(chain1[-(1:burnIn),]))
```

```
acceptance2 = 1-mean(duplicated(chain2[-(1:burnIn),]))
```

```
acceptance3 = 1-mean(duplicated(chain3[-(1:burnIn),]))
```

```
# Performance of the sampler: c = 0.1
```

```
par(mfrow=c(1,3))
```

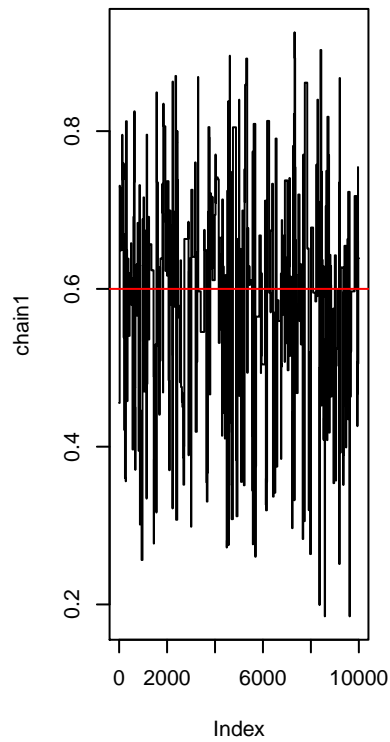
```
plot(chain1, type='l', main = "Trace plot of Chain, c = 0.1")
```

```
abline(h=0.6, col="red")
```

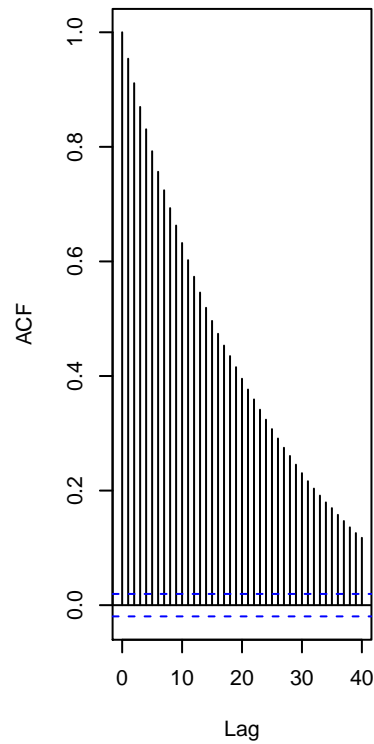
```
acf(chain1, main = "Autocorrelation plot of Chain, c= 0.1")
```

```
hist(chain1, xlab = "Samples from chain", main = "Histogram of Chain, c = 0.1")
```

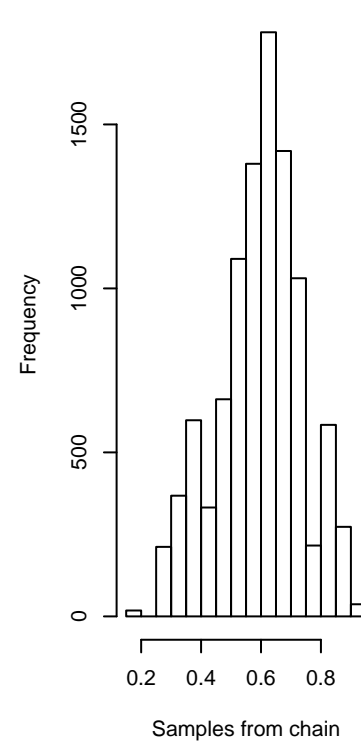
Trace plot of Chain, c = 0.1



Autocorrelation plot of Chain, c =

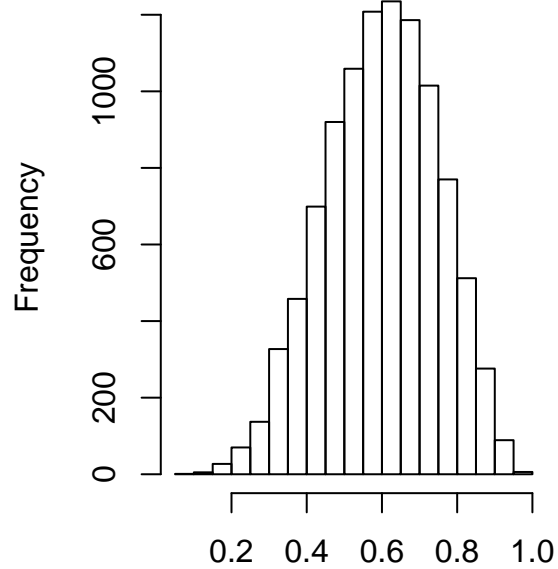


Histogram of Chain, c = 0.1



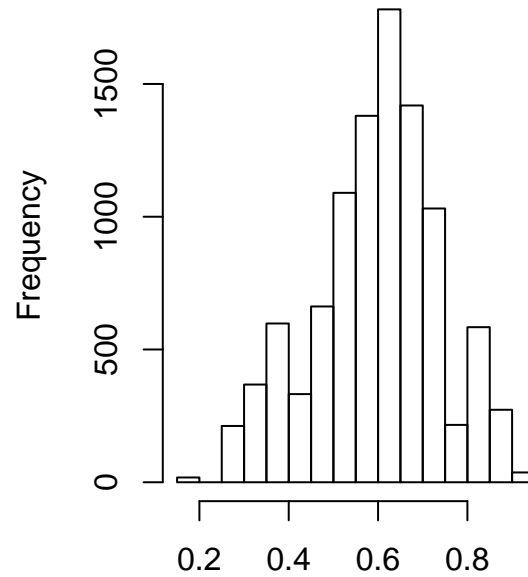
```
# Comparing the histogram of the chain with the target distribution Beta(6,4)
par(mfrow=c(1,2))
hist(test, xlab = "Samples from Beta(6, 4)", main = "Histogram of Beta(6, 4)")
hist(chain1, xlab = "Samples from chain", main = "Histogram of Chain, c = 0.1")
```

**Histogram of Beta(6, 4)**



**Samples from Beta(6, 4)**

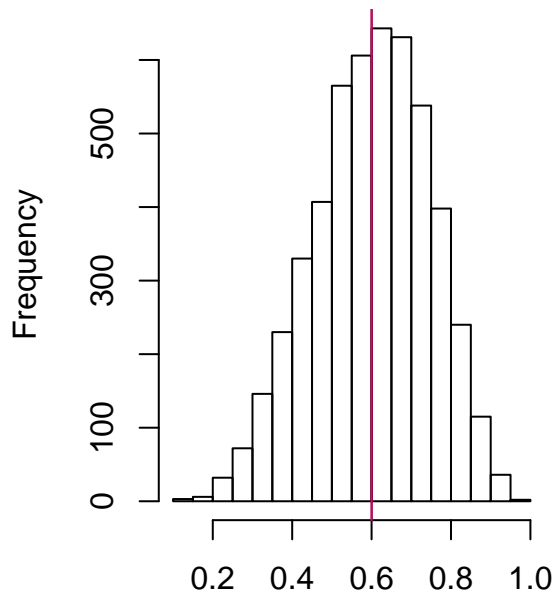
**Histogram of Chain, c = 0.1**



**Samples from chain**

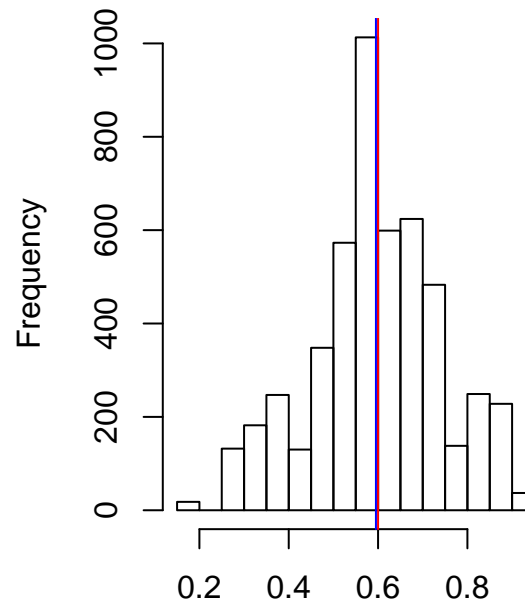
```
# Histogram of the target distribution Beta(6,4) vs. the Chain after burnIn
hist(test2, xlab = "Samples from Beta(6, 4)", main = "Histogram of Beta(6, 4)")
abline(v=mean(test2), col="blue")
abline(v=0.6, col="red")
hist(chain1[-(1:burnIn),1], xlab = "Samples from chain", main = "Histogram of Chain, c = 0.1")
abline(v = mean(chain1[-(1:burnIn),1]), col="blue")
abline(v=0.6, col="red")
```

### Histogram of Beta(6, 4)



Samples from Beta(6, 4)

### Histogram of Chain, c = 0.1



Samples from chain

```
# Kolmogorov-Smirnov statistic
```

```
ks.test(chain1[-(1:burnIn),1], test2)
```

```
## Warning in ks.test(chain1[-(1:burnIn), 1], test2): p-value will be
## approximate in the presence of ties
```

```
##
```

```
## Two-sample Kolmogorov-Smirnov test
```

```
##
```

```
## data: chain1[-(1:burnIn), 1] and test2
```

```
## D = 0.080848, p-value = 1.277e-14
```

```
## alternative hypothesis: two-sided
```

```
# Performance of the sampler: c = 2.5
```

```
par(mfrow=c(1,3))
```

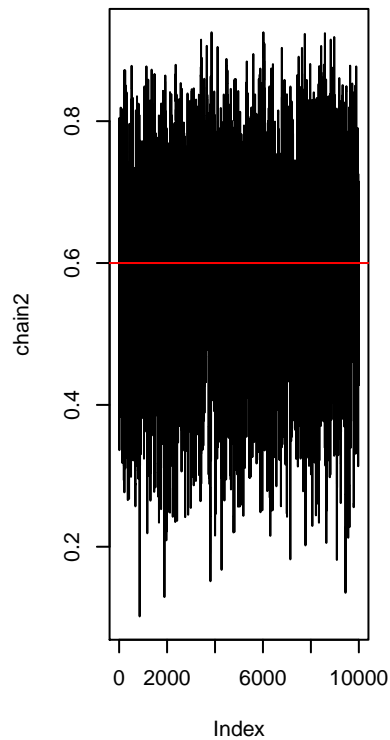
```
plot(chain2, type='l', main = "Trace plot of Chain, c = 2.5")
```

```
abline(h=0.6, col="red")
```

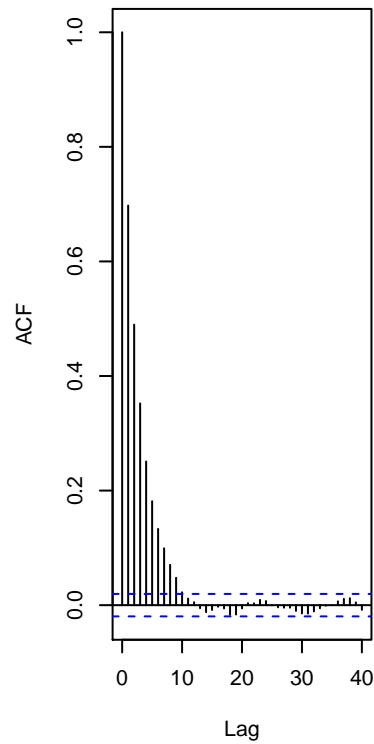
```
acf(chain2, main = "Autocorreltion plot of Chain, c = 2.5")
```

```
hist(chain2, xlab = "Samples from chain", main = "Histogram of Chain, c = 2.5")
```

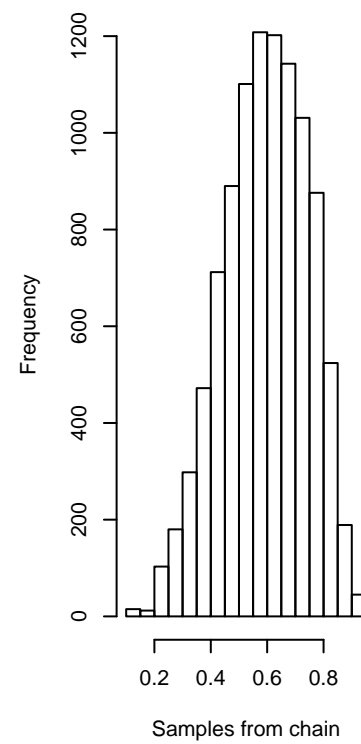
Trace plot of Chain, c = 2.5



Autocorrelation plot of Chain, c =



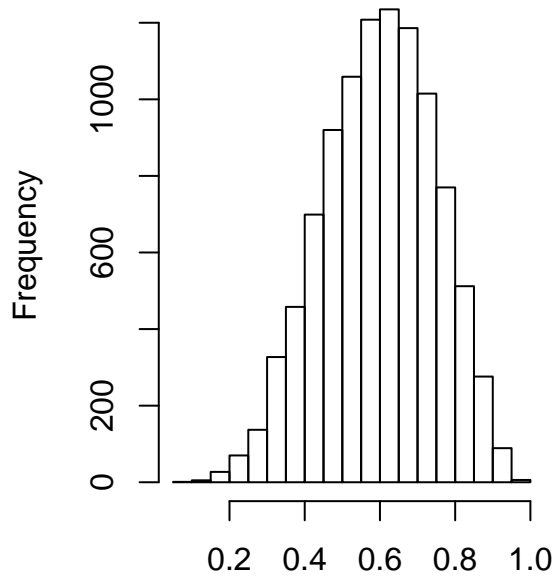
Histogram of Chain, c = 2.5



```
# Comparing the histogram of the chain with the target distribution Beta(6,4)
par(mfrow=c(1,2))
hist(test, xlab = "Samples from Beta(6, 4)", main = "Histogram of Beta(6, 4)")
hist(chain2, xlab = "Samples from chain", main = "Histogram of Chain, c = 2.5")
```

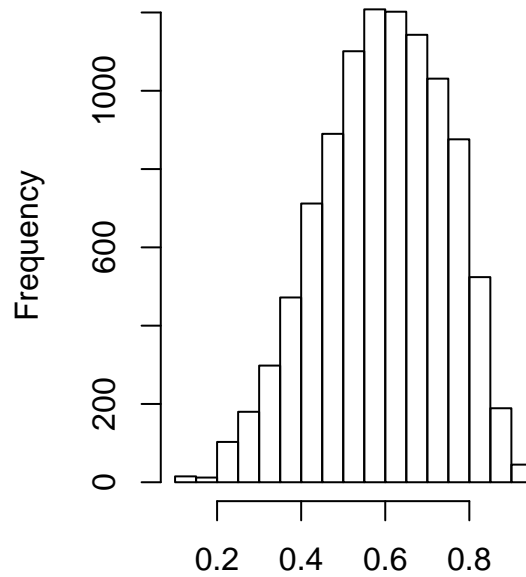


### Histogram of Beta(6, 4)



Samples from Beta(6, 4)

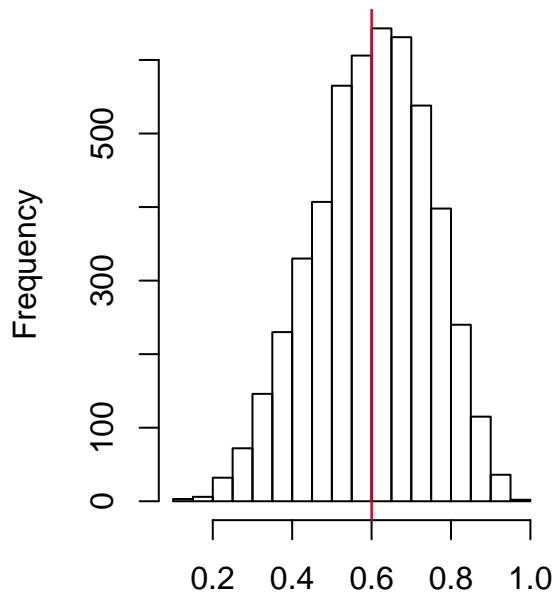
### Histogram of Chain, c = 2.5



Samples from chain

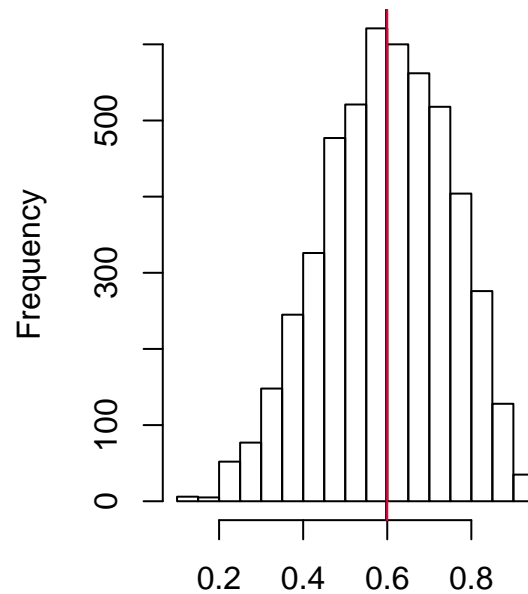
```
# Histogram of the target distribution Beta(6,4) vs. the Chain after burnIn
hist(test2, xlab = "Samples from Beta(6, 4)", main = "Histogram of Beta(6, 4)")
abline(v=mean(test2), col="blue")
abline(v=0.6, col="red")
hist(chain2[-(1:burnIn),1], xlab = "Samples from chain", main = "Histogram of Chain, c = 2.5")
abline(v = mean(chain2[-(1:burnIn),1]), col="blue")
abline(v=0.6, col="red")
```

**Histogram of Beta(6, 4)**



**Samples from Beta(6, 4)**

**Histogram of Chain, c = 2.5**



**Samples from chain**

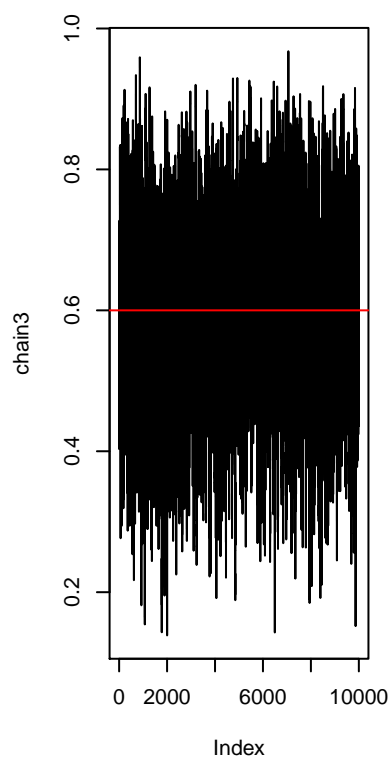
```
# Kolmogorov-Smirnov statistic  
ks.test(chain2[-(1:burnIn),1], test2)
```

```
## Warning in ks.test(chain2[-(1:burnIn), 1], test2): p-value will be  
## approximate in the presence of ties
```

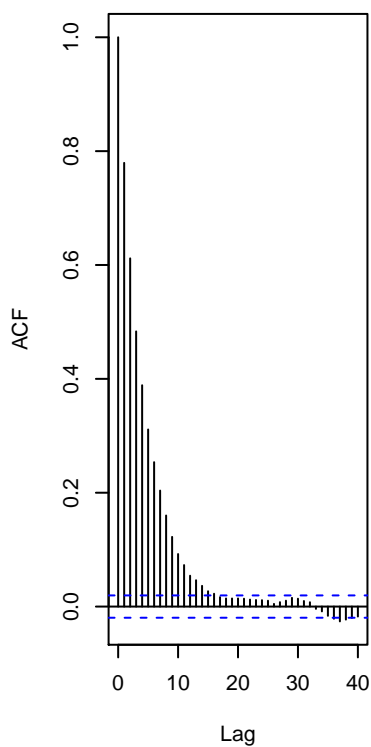
```
##  
## Two-sample Kolmogorov-Smirnov test  
##  
## data: chain2[-(1:burnIn), 1] and test2  
## D = 0.025304, p-value = 0.08138  
## alternative hypothesis: two-sided
```

```
# Performance of the sampler: c = 10  
par(mfrow=c(1,3))  
plot(chain3, type='l', main = "Trace plot of Chain, c = 10")  
abline(h=0.6, col="red")  
acf(chain3, main = "Autocorrelation plot of Chain, c = 10")  
hist(chain3, xlab = "Samples from chain", main = "Histogram of Chain, c = 10")
```

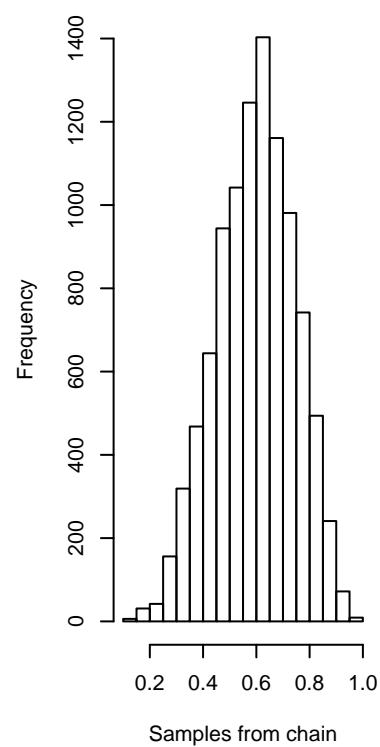
Trace plot of Chain, c = 10



Autocorrelation plot of Chain, c =

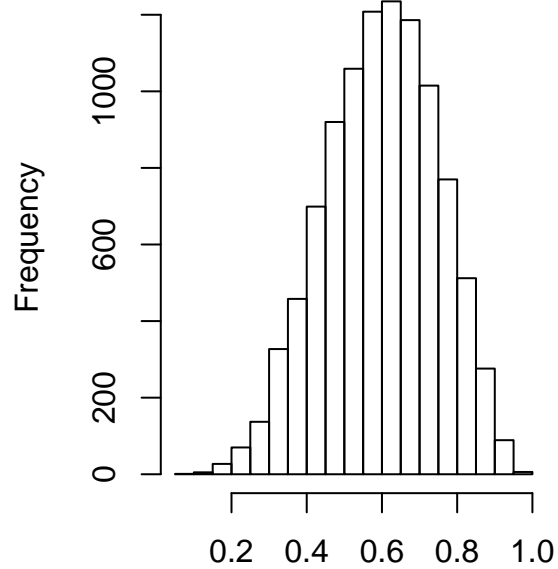


Histogram of Chain, c = 10



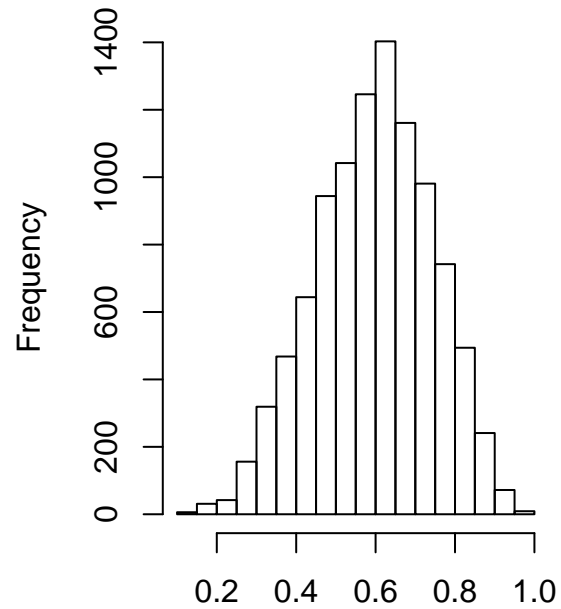
```
# Comparing the histogram of the chain with the target distribution Beta(6,4)
par(mfrow=c(1,2))
hist(test, xlab = "Samples from Beta(6, 4)", main = "Histogram of Beta(6, 4)")
hist(chain3, xlab = "Samples from chain", main = "Histogram of Chain, c = 10")
```

**Histogram of Beta(6, 4)**



**Samples from Beta(6, 4)**

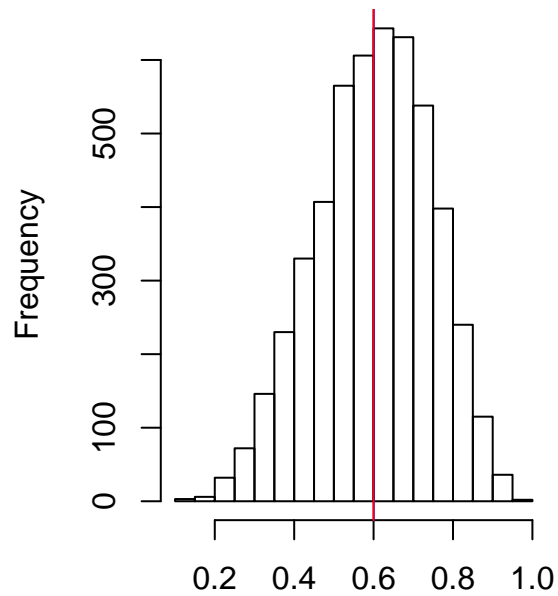
**Histogram of Chain, c = 10**



**Samples from chain**

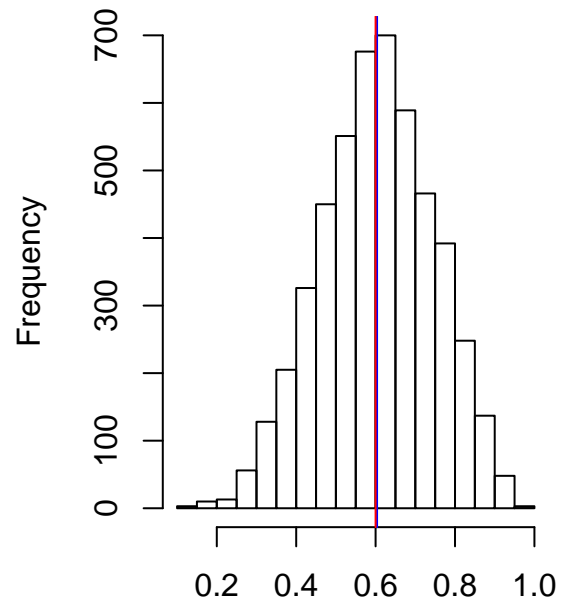
```
# Histogram of the target distribution Beta(6,4) vs. the Chain after burnIn
hist(test2, xlab = "Samples from Beta(6, 4)", main = "Histogram of Beta(6, 4)")
abline(v=mean(test2), col="blue")
abline(v=0.6, col="red")
hist(chain3[-(1:burnIn),1], xlab = "Samples from chain", main = "Histogram of Chain, c = 10")
abline(v = mean(chain3[-(1:burnIn),1]), col="blue")
abline(v=0.6, col="red")
```

**Histogram of Beta(6, 4)**



**Samples from Beta(6, 4)**

**Histogram of Chain, c = 10**



**Samples from chain**

```
# Kolmogorov-Smirnov statistic
ks.test(chain3[-(1:burnIn),1], test2)

## Warning in ks.test(chain3[-(1:burnIn), 1], test2): p-value will be
## approximate in the presence of ties

##
## Two-sample Kolmogorov-Smirnov test
##
## data: chain3[-(1:burnIn), 1] and test2
## D = 0.022024, p-value = 0.1768
## alternative hypothesis: two-sided
```