Final_Project_1

1. algorithm and implementation

- Functions

Target function is a beta distribution with shape1 = 6 and shape2 = 4 Proposal function generates a ϕ_{new} from a beta distribution with shape1 = cϕ_{old} and shape2 = c(1 - ϕ_{old})

```
target <- function(x){
  if ((x > 1) | (x < 0)){
    return(0)}
  else{
    return(dbeta(x, 6, 4))
  }
}

proposalfunction <- function(c, shape){

  return(rbeta(1, c * shape, c * (1 - shape)))
}
```

- Algorithm

let t(x) stand for the target function

let p(x) stand for the proposal function

let q(x) stand for the acceptance rate funtion

Given x_i, generate Y_i ~ p(y|x_i)

Then the acceptance probability for $X_{i+1} = Y_i$ is q(x).

The probability for $X_{i+1} = X_i$ is 1 - q(x).

Since beta distribution is asymmetric, we need to implement a correction factor when calculating the acceptance probability.

acceptance probability $= \frac{t(p(x_i))}{t(x_i)} \times correction$

where correction $= \frac{\phi(x_i, c\times p(x_i), c\times(1-p(x_i))}{\phi(p(x_i), c\times p(x_i), c\times(1-p(x_i))}$

when p(x) >= 1, then jump to the new Y_i as well.

- Code

```
run_metropolis_MCMC <- function(c, startvalue, iterations){
    chain <- rep(0, iterations)
    chain[1] <- startvalue
    for (i in 1:iterations){
        #generate candidate Y_{i} using proposal function
        proposal <- proposalfunction(c, chain[i])
        #calculate the correction factor
        correction <- dbeta(chain[i], c * proposal, c * (1- proposal))/ dbeta(proposal, c * proposal, c
        #calculate the acceptance rate
        probab <- (target(proposal) / target(chain[i]) )* correction
        #generate a uniform random value
```

```
        #and compare it with the acceptance rate.
        #If acceptance rate >= this value,
        #then we have X_{i+1} = Y_{i}.
        if (runif(1) <= probab){
            chain[i+1] = proposal
        }else{
          #Otherwise, we keep x_{i} as the new X_{i+1} = x_{}
            chain[i+1] = chain[i]
        }
    }
    return(chain)
}
```

2. initial run: c = 1

c = 1, initial runs = 10000, generate a start value from a uniform distribution on [0, 1], with no burn-in time.

From the histogram we can see that randomly generated values from the target is less centered than the sample.

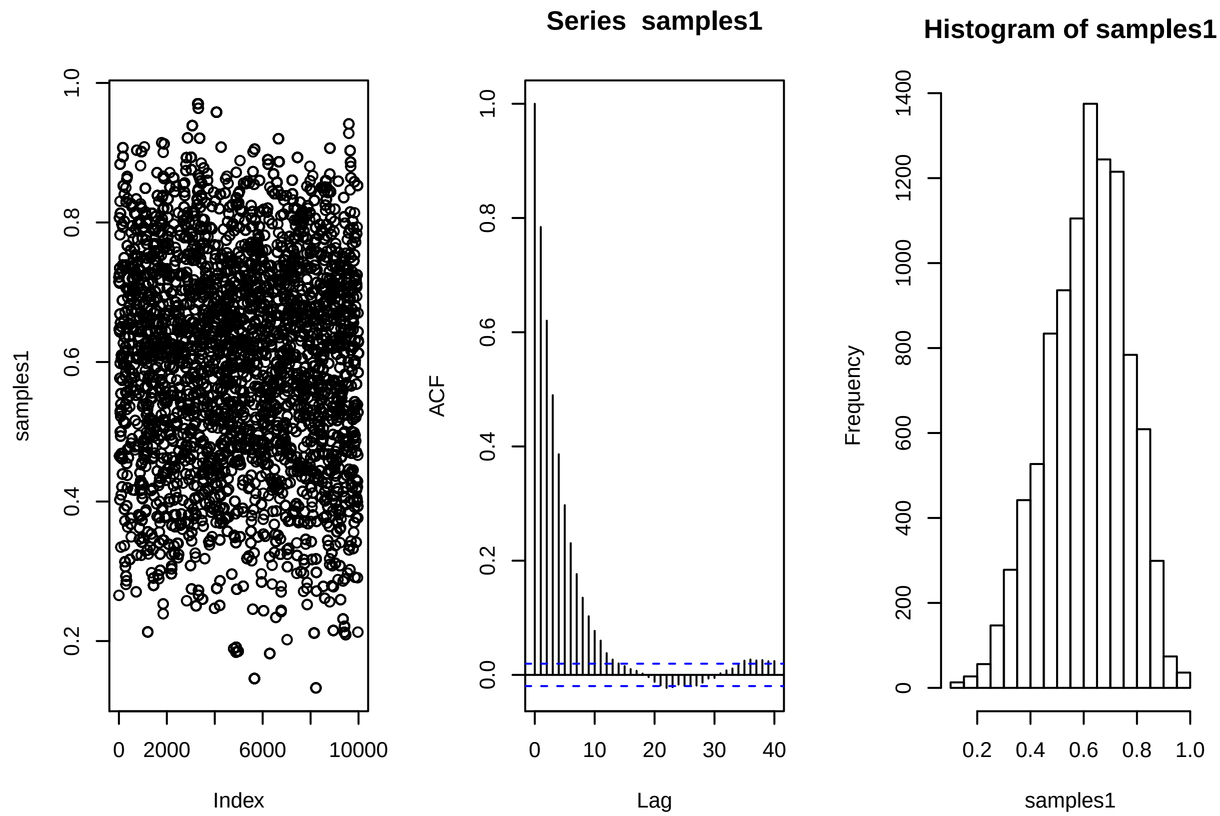
In ks-test, D-value is greater than critical value at a confidence level of 0.95.

```
set.seed(1)
start <- runif(1, 0, 1)
samples1 <- run_metropolis_MCMC(1, start, 10000)
acceptance_rate1 <- 1 - mean(duplicated(samples1))
print(c("acceptance rate is: ", acceptance_rate1))
```

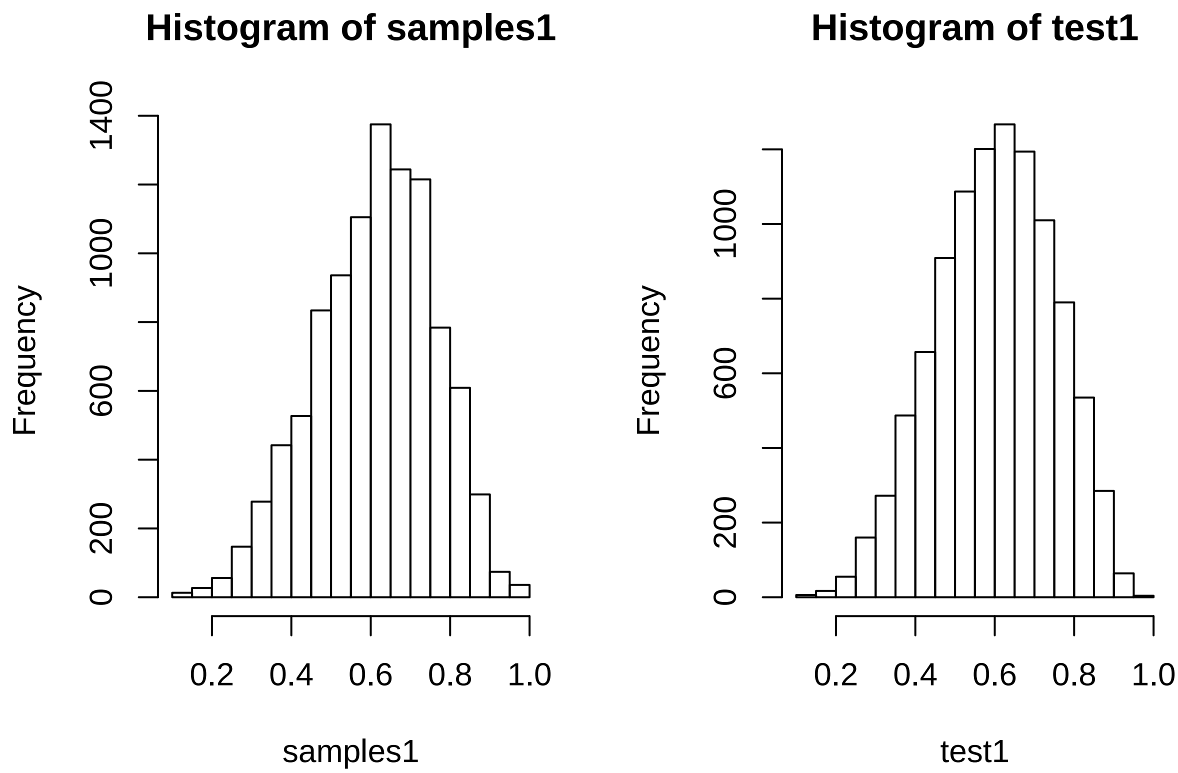
```
## [1] "acceptance rate is: " "0.257374262573743"
```

Provide a trace plot of this sampler and an autocorrelation plot, as well as a histogram of the draws.

```
par(mfrow=c(1,3))  #1 row, 3 columns
plot(samples1); acf(samples1); hist(samples1)  #plot commands
```




```
#compare with beta distribution with shape 1 = 6 and shape 2 = 4
test1 = rbeta(10000,6,4)
par(mfrow=c(1,2))
hist(samples1)
hist(test1)
```



- the Kolmogorov–Smirnov statistic

```r
#install.packages("BoutrosLab.plotting.general")
library(BoutrosLab.plotting.general)
```

```
## Warning: package 'BoutrosLab.plotting.general' was built under R version
## 3.4.4

## Loading required package: lattice

## Loading required package: latticeExtra

## Loading required package: RColorBrewer

## Loading required package: cluster

## Loading required package: hexbin

## Warning: package 'hexbin' was built under R version 3.4.3

## Loading required package: grid

##
## Attaching package: 'BoutrosLab.plotting.general'

## The following object is masked from 'package:stats':
##
##     dist
```

```r
ks.test.critical.value(10000, 0.95)
```

```
## [1] 0.013581
```

```r
ks.test(samples1, pbeta, 6, 4)
```

```
## Warning in ks.test(samples1, pbeta, 6, 4): ties should not be present for
## the Kolmogorov-Smirnov test
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  samples1
## D = 0.05242, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

3. runs

(a) without adding burn-in time

(1) c = 0.1

When c = 0.1, acceptance rate ~ 0.04.

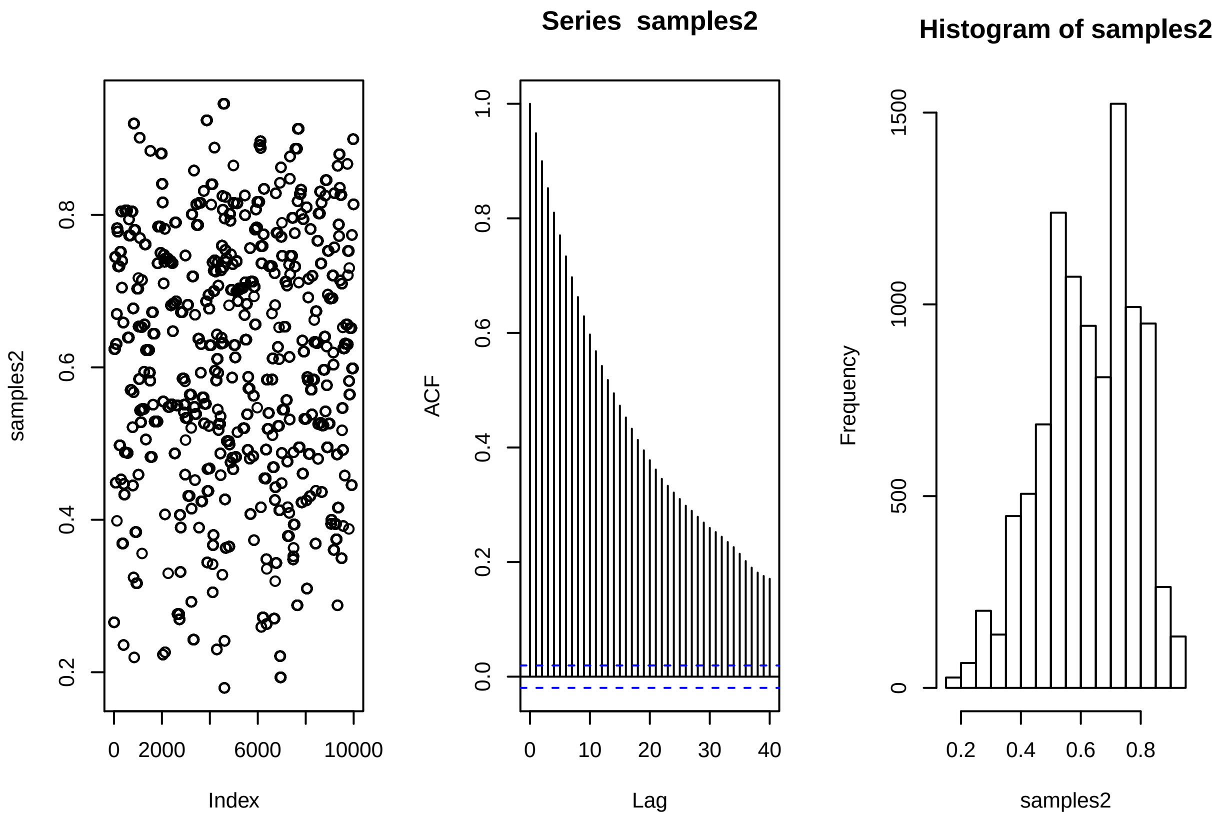
From the histogram we can see that the sample does not have the same shape as the randomly generated values from the target.

In ks-test, D-value is smaller than critical value at a confidence level of 0.95.

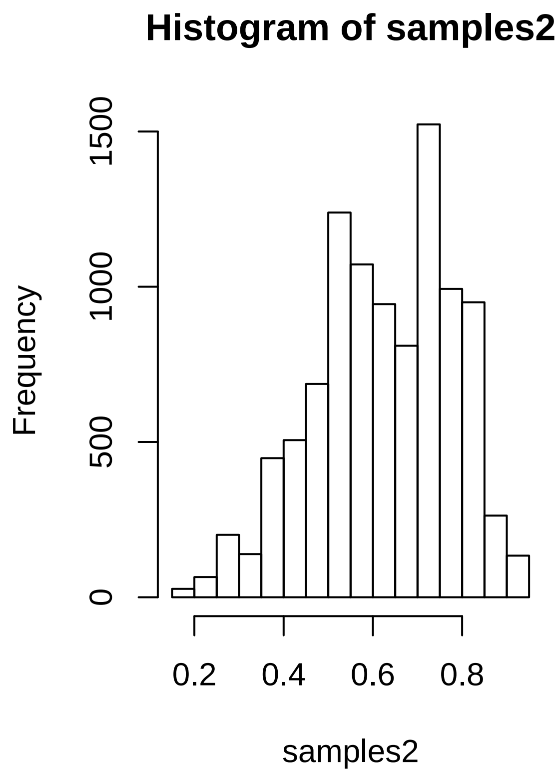
```r
samples2 <- run_metropolis_MCMC(0.1, start, 10000)
acceptance_rate2 <- 1 - mean(duplicated(samples2))
print(acceptance_rate2)
```

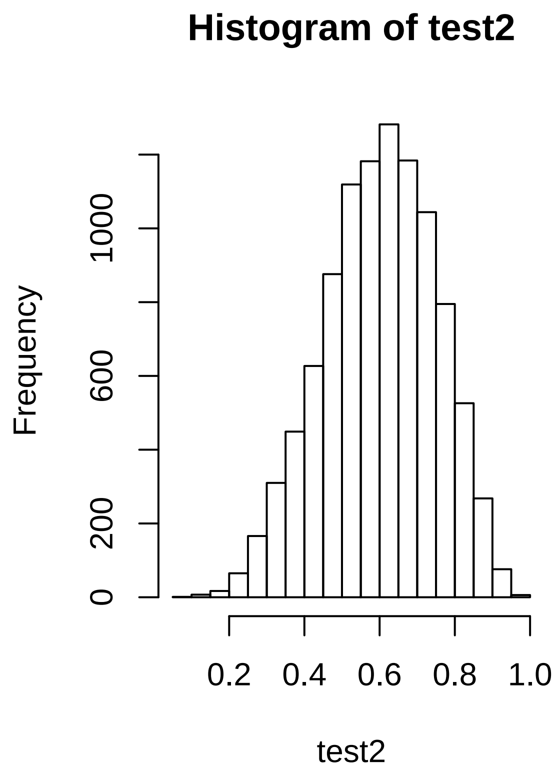
```
## [1] 0.04159584
```

```r
 par(mfrow=c(1,3))  #1 row, 3 columns
 plot(samples2); acf(samples2); hist(samples2)  #plot commands
```




```
#compare with beta distribution with shape 1 = 6 and shape 2 = 4
test2 = rbeta(10000,6,4)
par(mfrow=c(1,2))
hist(samples2)
hist(test2)
```





- the Kolmogorov–Smirnov statistic

```
ks.test(samples2, pbeta, 6, 4)
```

```
## Warning in ks.test(samples2, pbeta, 6, 4): ties should not be present for
## the Kolmogorov-Smirnov test

##
##  One-sample Kolmogorov-Smirnov test
##
## data:  samples2
## D = 0.11709, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

(2) c = 2.5

When c = 2.5, acceptance rate ~ 0.4.

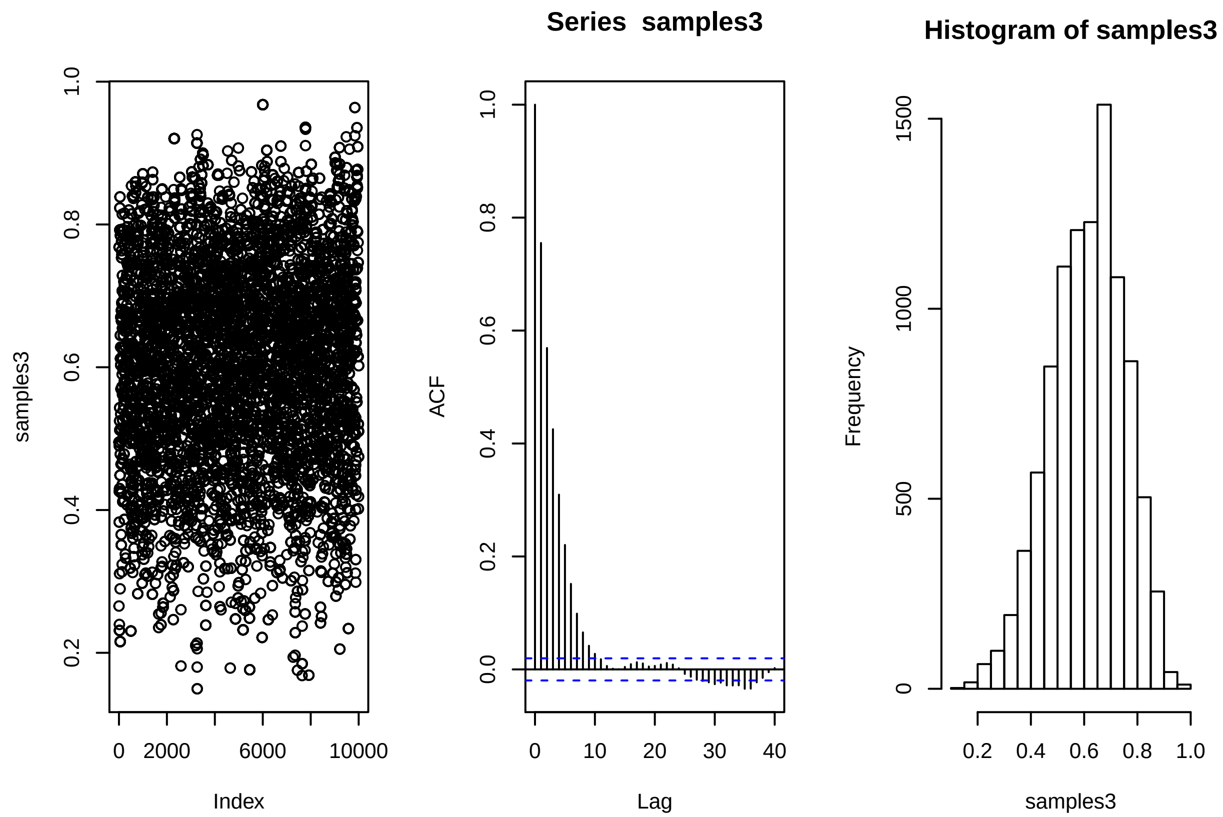
From the histogram we can see that the sample has a closer shape to the randomly generated values from the target.

In ks-test, D-value is greater than critical value at a confidence level of 0.95.

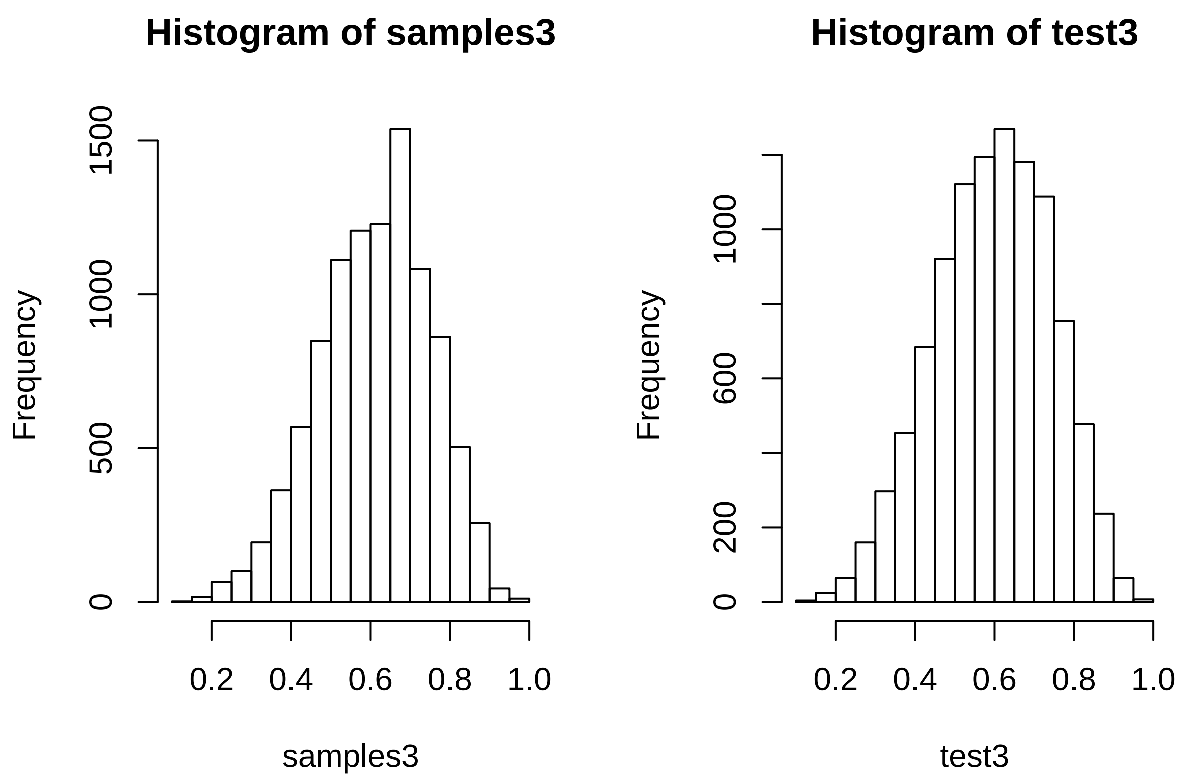
```
samples3 <- run_metropolis_MCMC(2.5, start, 10000)
acceptance_rate3 <- 1 - mean(duplicated(samples3))
print(acceptance_rate3)
```

```
## [1] 0.3936606
```

```r
par(mfrow=c(1,3))   #1 row, 3 columns
plot(samples3); acf(samples3); hist(samples3)   #plot commands
```




```r
#compare with beta distribution with shape 1 = 6 and shape 2 = 4
test3 = rbeta(10000,6,4)
par(mfrow=c(1,2))
hist(samples3)
hist(test3)
```



- the Kolmogorov–Smirnov statistic

```
ks.test(samples3, pbeta, 6, 4)
```

```
## Warning in ks.test(samples3, pbeta, 6, 4): ties should not be present for
## the Kolmogorov-Smirnov test
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  samples3
## D = 0.041799, p-value = 1.332e-15
## alternative hypothesis: two-sided
```

(3) c = 10

When c = 2.5, acceptance rate ~ 0.4.

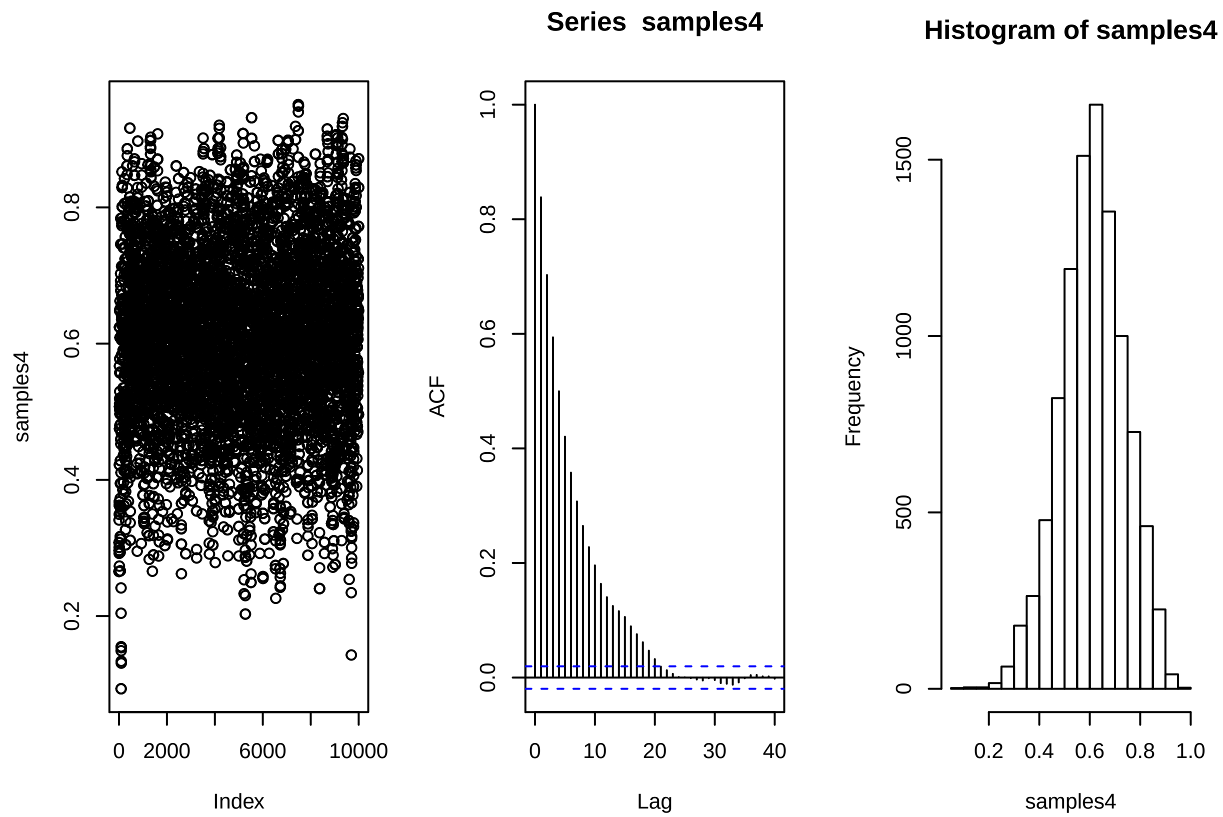
From the histogram we can see that the sample has a narrower shape than the randomly generated values from the target.

In ks-test, D-value is greater than critical value at a confidence level of 0.95.

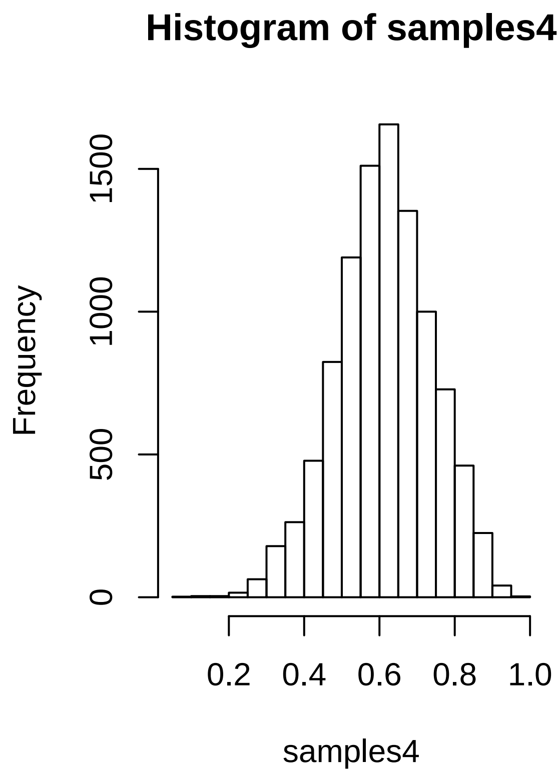
```
samples4 <- run_metropolis_MCMC(10, start, 10000)
acceptance_rate4 <- 1 - mean(duplicated(samples4))
print(acceptance_rate4)
```

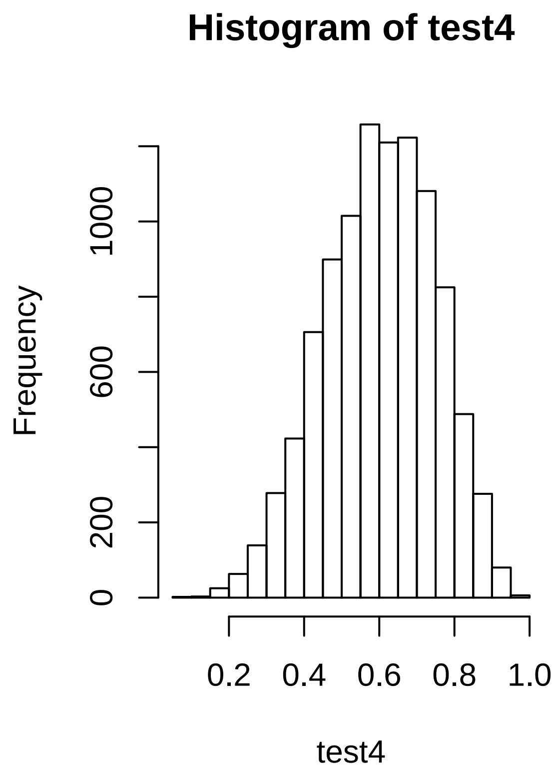
```
## [1] 0.5811419
```

```r
par(mfrow=c(1,3))  #1 row, 3 columns
plot(samples4); acf(samples4); hist(samples4)  #plot commands
```




```r
#compare with beta distribution with shape 1 = 6 and shape 2 = 4
test4 = rbeta(10000,6,4)
par(mfrow=c(1,2))
hist(samples4)
hist(test4)
```





- the Kolmogorov–Smirnov statistic

```
ks.test(samples4, pbeta, 6, 4)
```

```
## Warning in ks.test(samples4, pbeta, 6, 4): ties should not be present for
## the Kolmogorov-Smirnov test

##
##  One-sample Kolmogorov-Smirnov test
##
## data:  samples4
## D = 0.071777, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

(2) adding burn-in time

Since acceptance rate for c = 0.1 is too low, it should have a later burn-in time than c = 2.5 and c = 10. acceptance for c = 10 is the highest among the three values, D-value for c = 2.5 is the smallest.

Therefore, we expect a smaller burn-in value for c = 2.5 and c = 10.

```
burnIn2 <- 0.7 * 10000
burnIn3 <- 0.5 * 10000
burnIn4 <- 0.3 * 10000
draws1 = samples1[-(1:burnIn3)]
draws2 = samples2[-(1:burnIn2)]
draws3 = samples3[-(1:burnIn3)]
draws4 = samples4[-(1:burnIn4)]
```

```r
acceptance1 <- 1 - mean(duplicated(draws1))
acceptance2 <- 1 - mean(duplicated(draws2))
acceptance3 <- 1 - mean(duplicated(draws3))
acceptance4 <- 1 - mean(duplicated(draws4))
```

```r
print(c(acceptance2, acceptance3, acceptance4))
```

```
## [1] 0.04331889 0.39092182 0.57820311
```

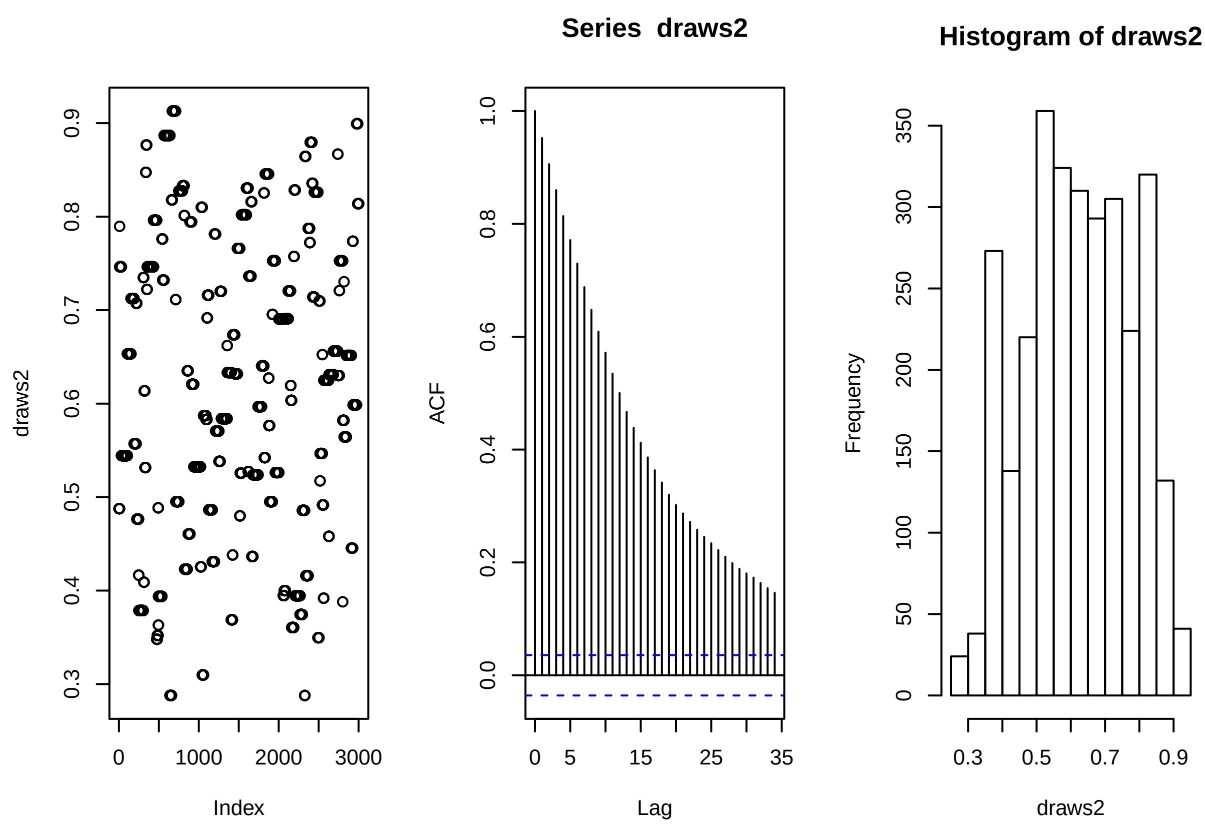
```r
print(c(1 - mean(duplicated(samples3[-(1:3000)])), 1 - mean(duplicated(samples3[-(1:5000)])), 1 - mean(
```

```
## [1] 0.3939437 0.3909218 0.3813093
```

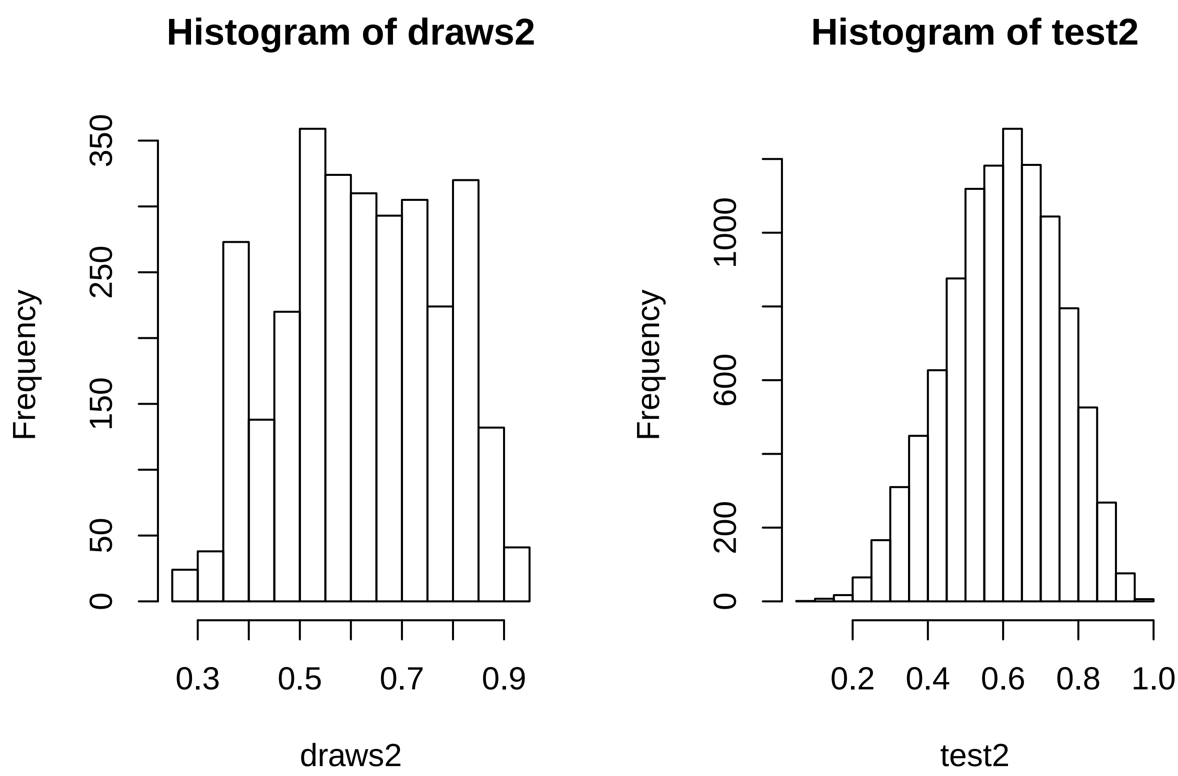
We can see that after adding a burn-in time, the acceptance rate for c = 2.5 and c = 10 does not change much.

(b) c = 2.5

```r
par(mfrow=c(1,3))  #1 row, 3 columns
plot(draws2); acf(draws2); hist(draws2)  #plot commands
```




```r
#compare with beta distribution with shape 1 = 6 and shape 2 = 4
test3 = rbeta(300,6,4)
par(mfrow=c(1,2))
hist(draws2)
hist(test2)
```



- the Kolmogorov–Smirnov statistic

```
print(ks.test.critical.value(3000, 0.95))
```

```
## [1] 0.0247954
```

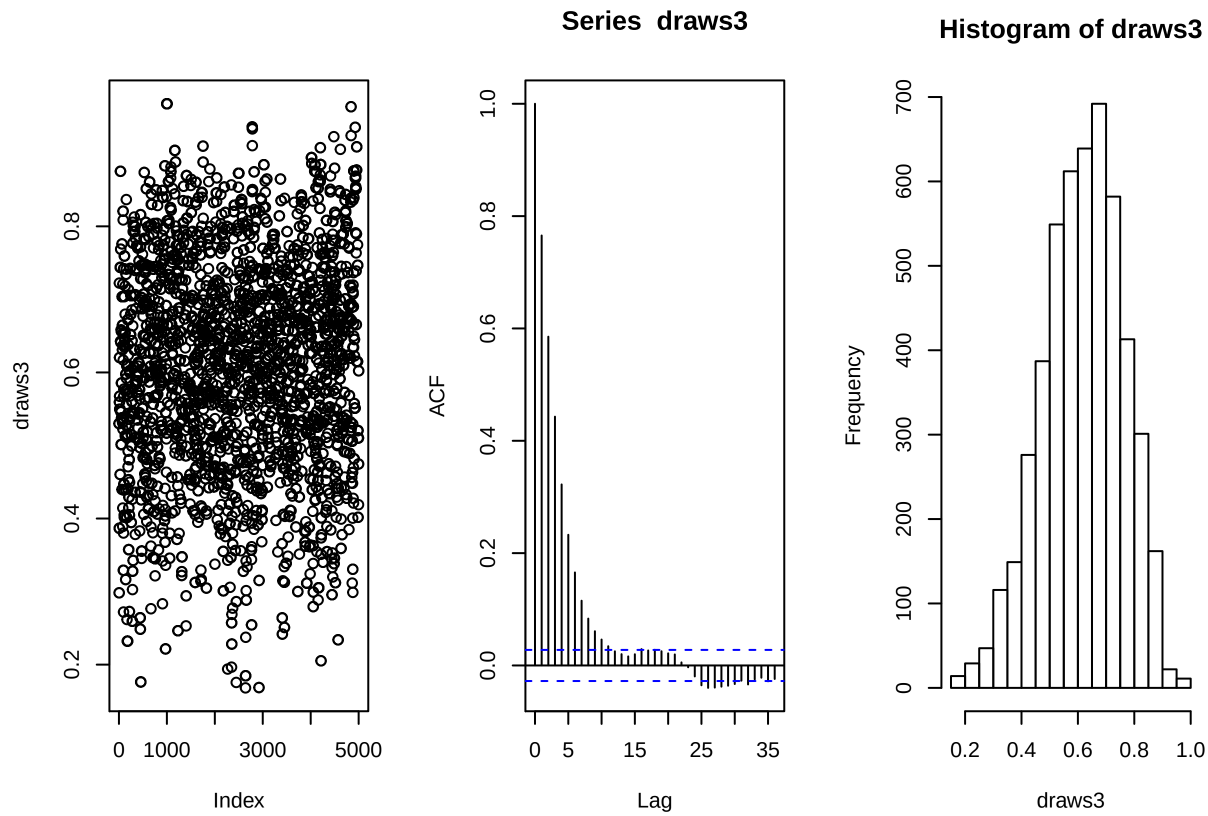
```
ks.test(draws2, pbeta, 6, 4)
```

```
## Warning in ks.test(draws2, pbeta, 6, 4): ties should not be present for the
## Kolmogorov-Smirnov test
```

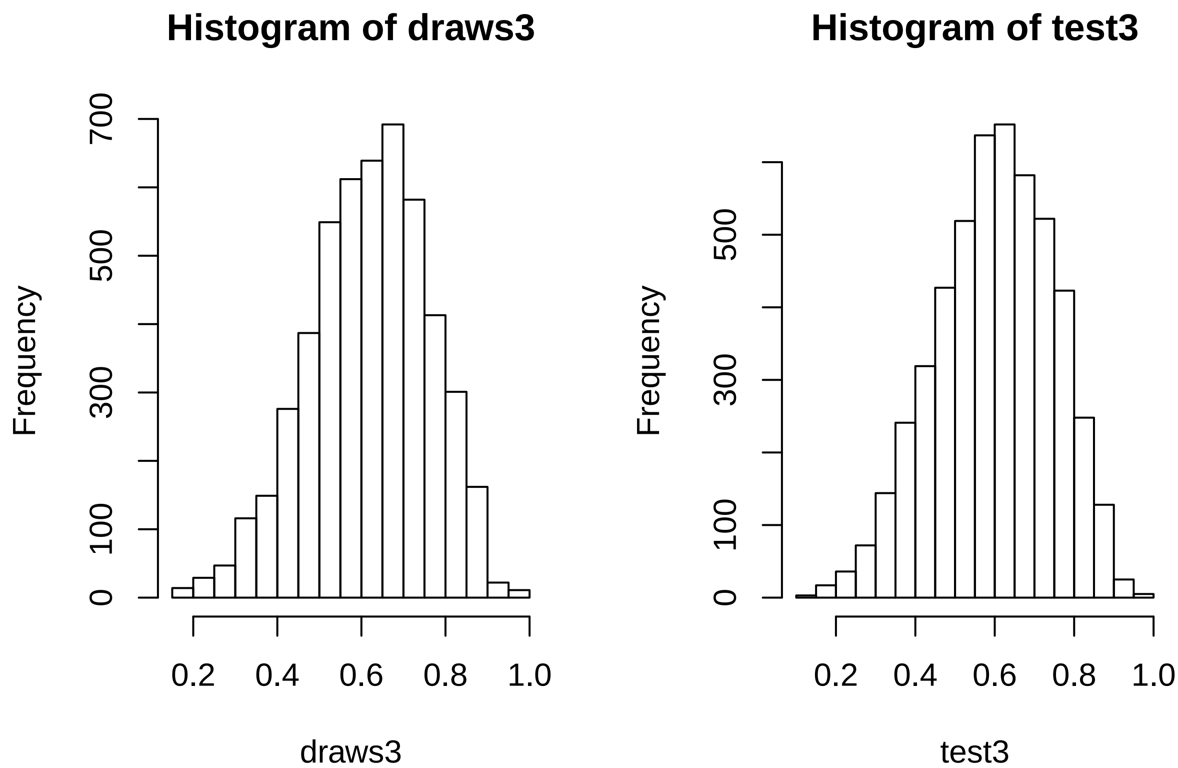
```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  draws2
## D = 0.099896, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

(b) c = 2.5

```
par(mfrow=c(1,3))   #1 row, 3 columns
plot(draws3); acf(draws3); hist(draws3)   #plot commands
```




```
#compare with beta distribution with shape 1 = 6 and shape 2 = 4
test3 = rbeta(5000,6,4)
par(mfrow=c(1,2))
hist(draws3)
hist(test3)
```



- the Kolmogorov–Smirnov statistic

```
print(ks.test.critical.value(5000, 0.95))
```

```
## [1] 0.01920643
```

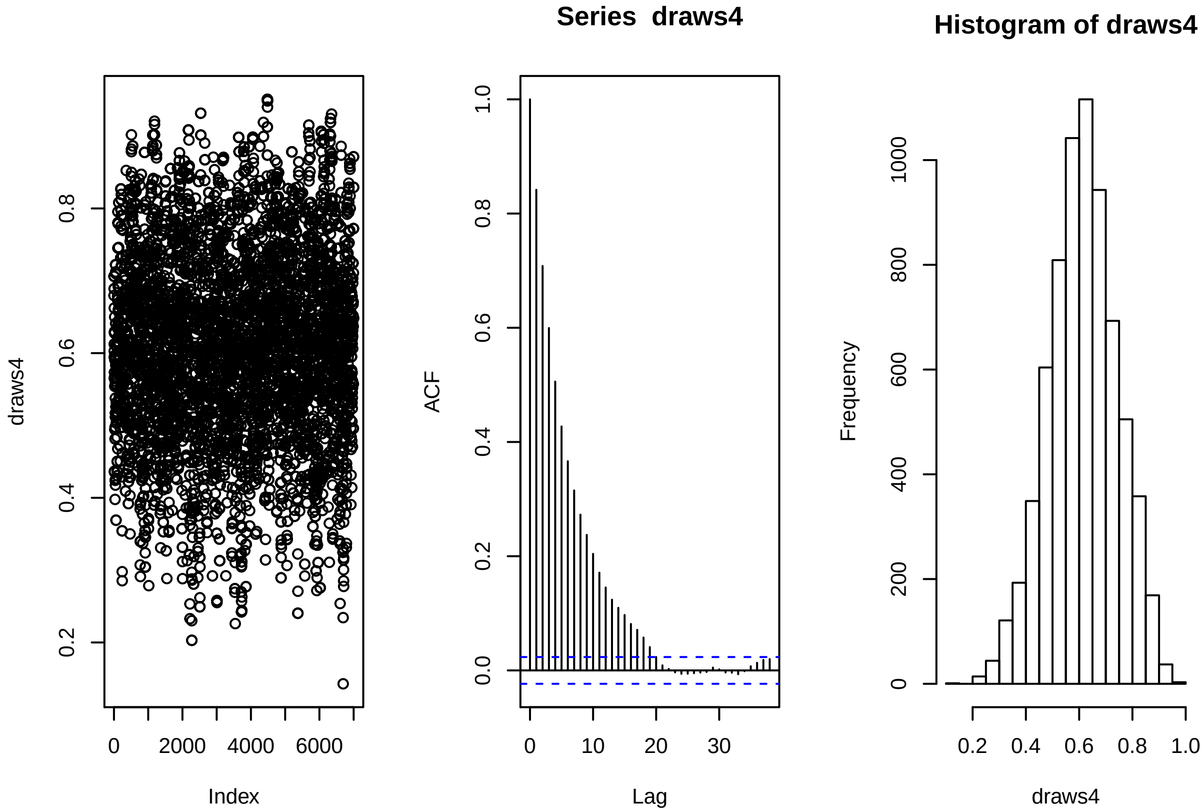
```
ks.test(draws3, pbeta, 6, 4)
```

```
## Warning in ks.test(draws3, pbeta, 6, 4): ties should not be present for the
## Kolmogorov-Smirnov test
```

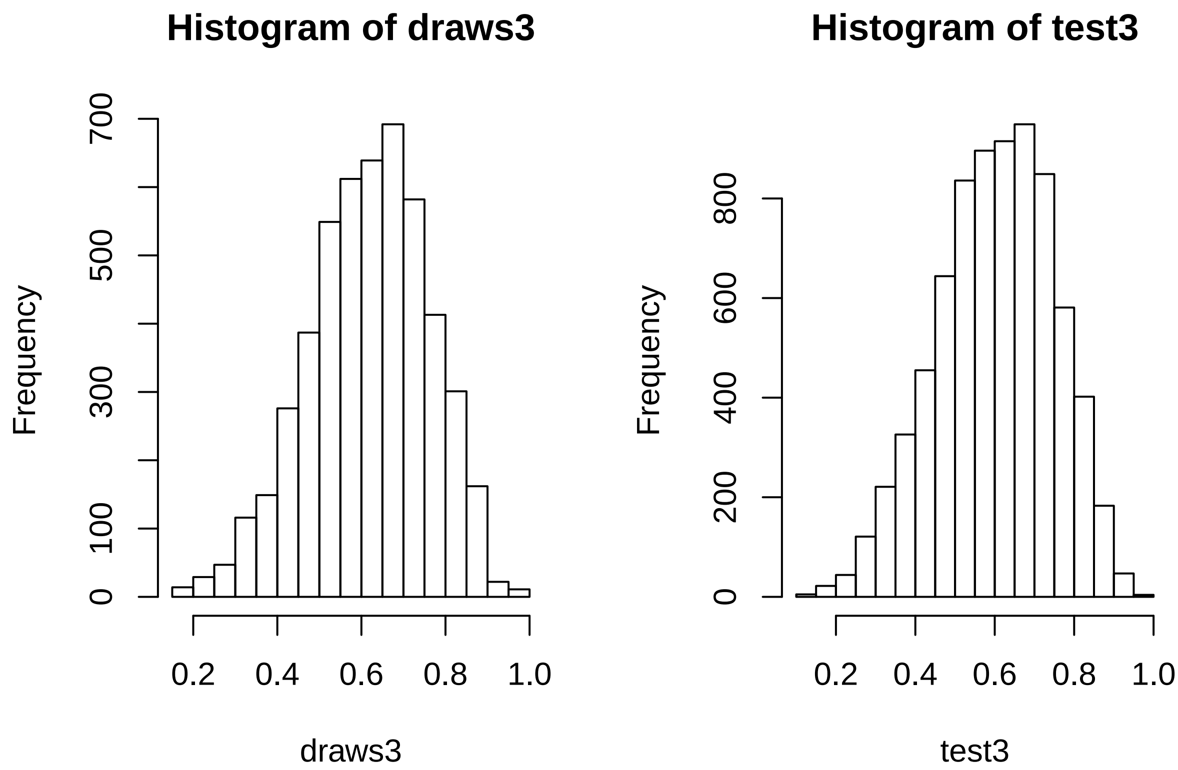
```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  draws3
## D = 0.05426, p-value = 3.253e-13
## alternative hypothesis: two-sided
```

(c) $c = 10$

```
par(mfrow=c(1,3))  #1 row, 3 columns
plot(draws4); acf(draws4); hist(draws4)  #plot commands
```




```r
#compare with beta distribution with shape 1 = 6 and shape 2 = 4
test3 = rbeta(7500,6,4)
par(mfrow=c(1,2))
hist(draws3)
hist(test3)
```



- the Kolmogorov–Smirnov statistic

```
print(ks.test.critical.value(7000, 0.95))
```

```
## [1] 0.0162324
```

```
ks.test(draws4, pbeta, 6, 4)
```

```
## Warning in ks.test(draws4, pbeta, 6, 4): ties should not be present for the
## Kolmogorov-Smirnov test
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  draws4
## D = 0.069267, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

Based on histograms and D-value from KS-test, $c = 2.5$ has the most similar shape to beta(6,4) and the smallest difference between D-Value and the critical value.