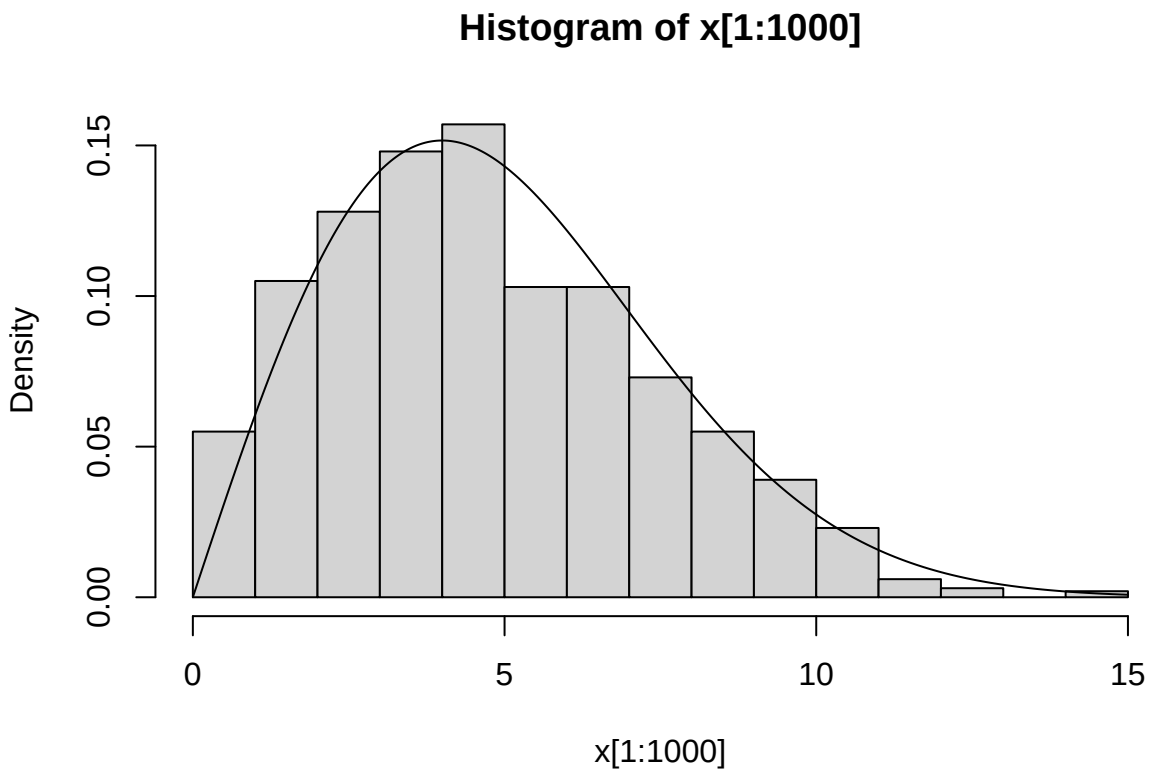
statHW5

Question 1

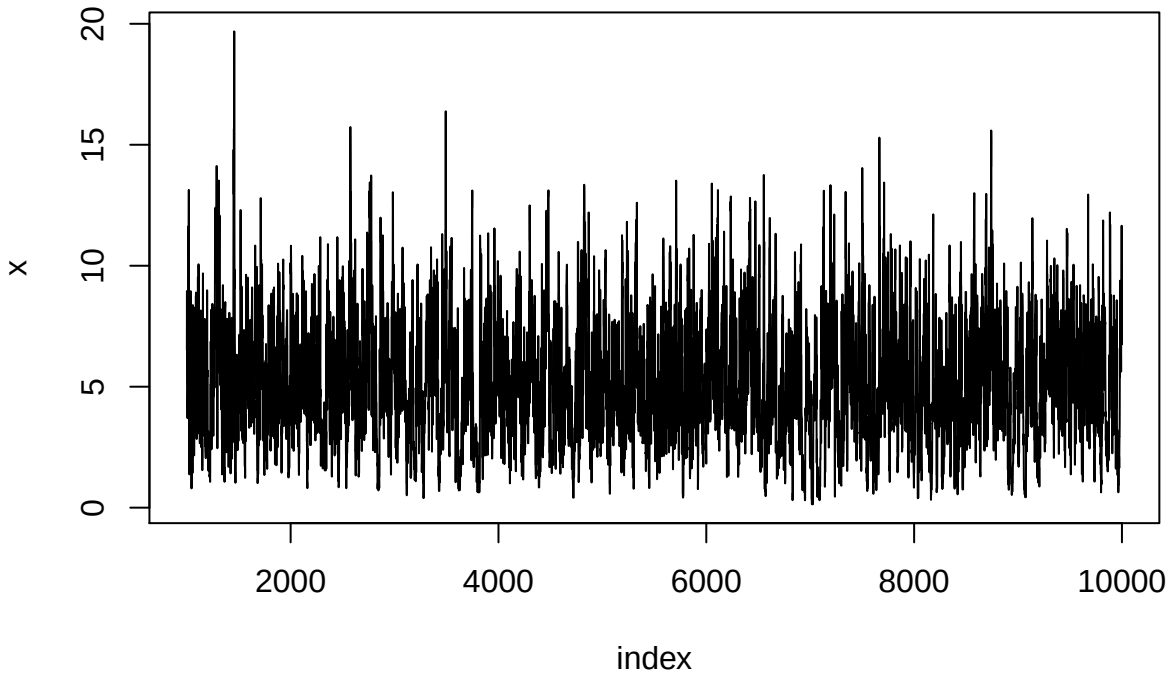
```
f = function(x, sigma=4) {
  if (any(x < 0)) return (0)
  stopifnot(sigma > 0)
  return((x / sigma^2) * exp(-x^2 / (2*sigma^2)))
}

m = 10000
sigma = 4
x = numeric(m)
x[1] = rgamma(1, 1, 1)
k = 0
u = runif(m)

for (i in 2:m) {
  xt = x[i-1]
  y = rgamma(1, shape = xt, rate = 1)
  num = f(y, sigma) * dgamma(xt, shape = y, rate = 1)
  den = f(xt, sigma) * dgamma(y, shape = xt, rate = 1)
  if (u[i] <= num/den)
    x[i] = y
  else {
    x[i] = xt
    k = k+1
  }
}

#rejection rate
print(k/m)
```

```
## [1] 0.3087
```

```
#histogram
hist(x[1:1000], freq=F)
xx = seq(0,15,0.01)
lines(xx, f(xx,sigma))
```




```
#plot
b0 = 1000 #burn-in
index = (b0+1):m
y1 = x[index]
plot( y1~index, type="l", main="", ylab="x" )
```



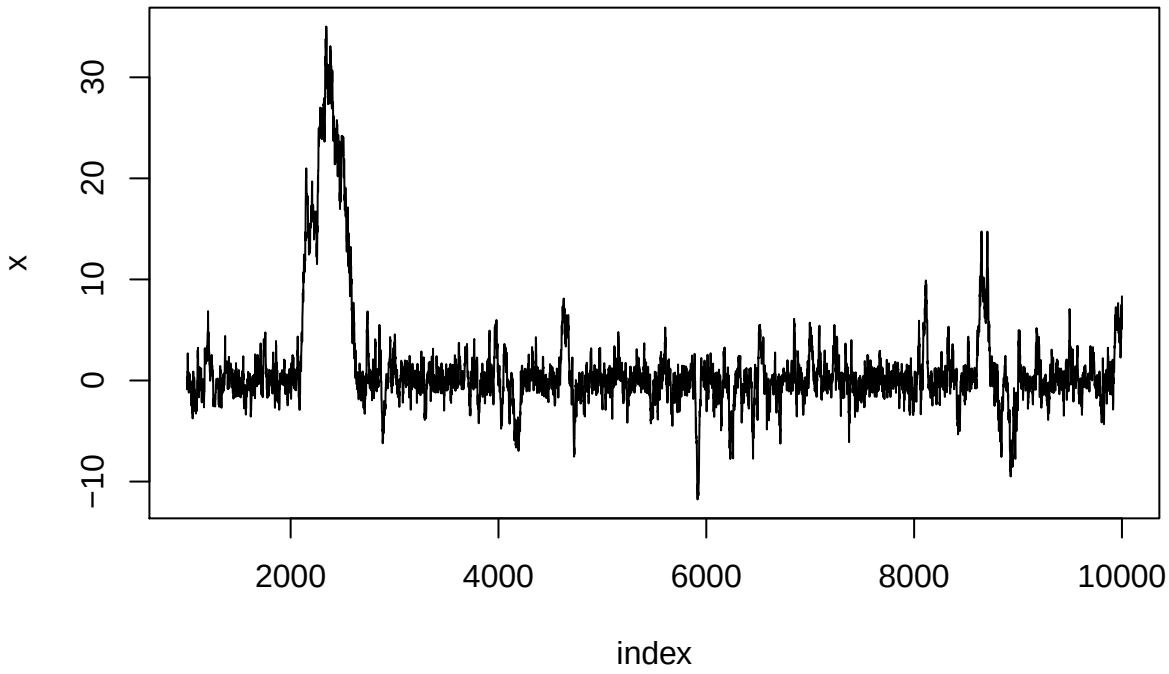
The histogram seems to agree with the target curve.

Question 2

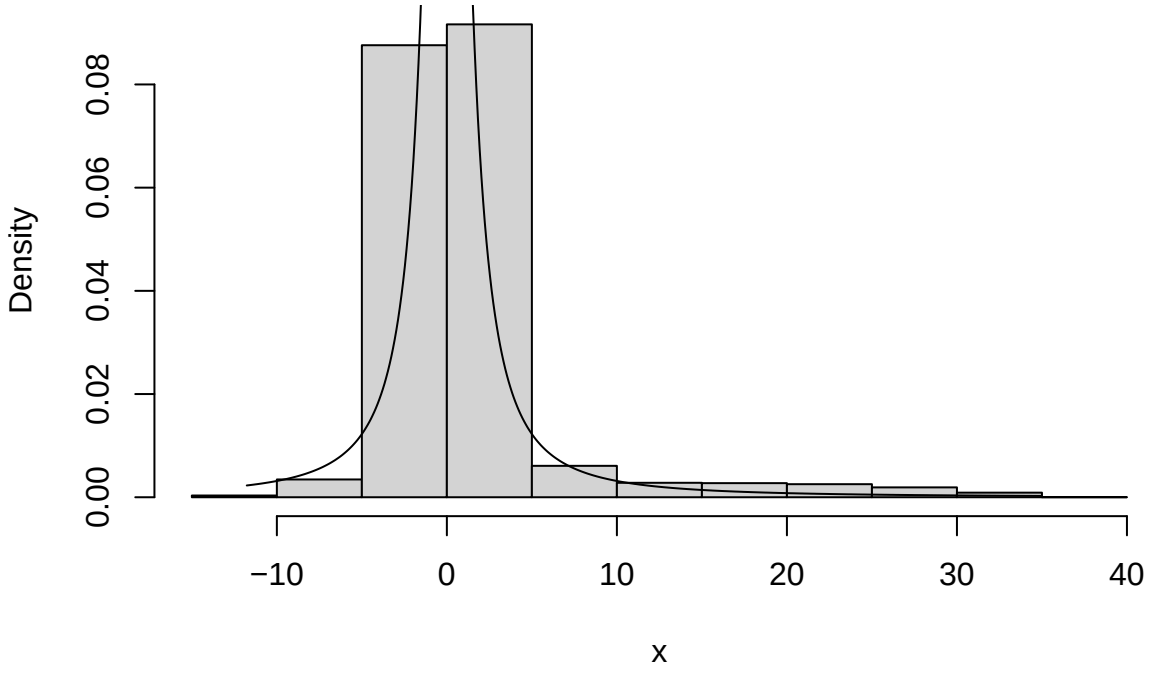
```
f2 = function(x, eta=0, theta=1) {
  stopifnot( theta > 0 )
  return( 1 / (pi*theta * (1 + ((x-eta)/theta)^2)) )
}
m = 10000
sigma = 1
x = numeric(m)
x[1] = rnorm(1,0,sigma)
k = 0
u = runif(m)
for (i in 2:m) {
  xt = x[i-1]
  y = rnorm(1, mean = xt, sd = sigma)
  num = f2(y) * dnorm(xt, mean = y, sd = sigma)
  den = f2(xt) * dnorm(y, mean = xt, sd = sigma)
  if (u[i] <= num/den) x[i] <- y
  else {
    x[i] = xt
    k = k+1
  }
}
#rejection rate
print(k/m)
```

```
## [1] 0.2268
```

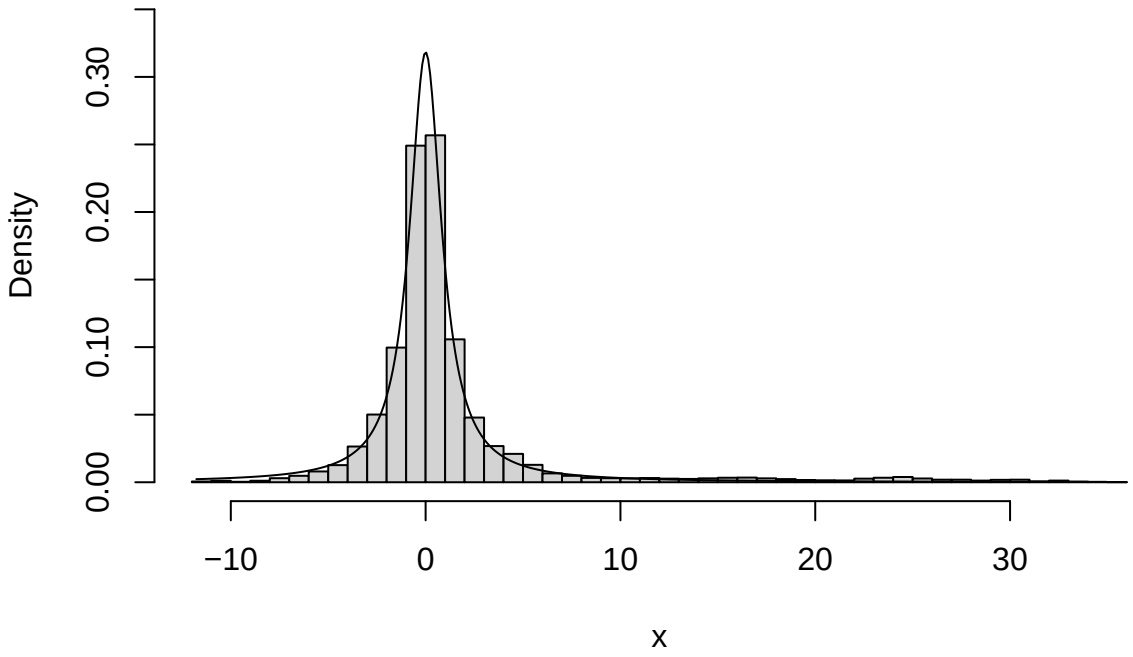
```
b0 = 1000 #burn-in
index = (b0+1):m
y1 = x[index]

plot( y1~index,  type="l", main="", ylab="x" )
```




```
hist( y1, prob=T, main='', xlab='x')
xx = seq( min(y1), max(y1), 0.1 )
lines( xx, f2(xx))
```




```
#neater plot
hist( y1, prob=T, main='', xlab='x', ylim=c(0,.35), breaks=50 )
xx = seq( min(y1), max(y1), 0.1 )
lines( xx, f2(xx), ylim=c(0,.35) )
```

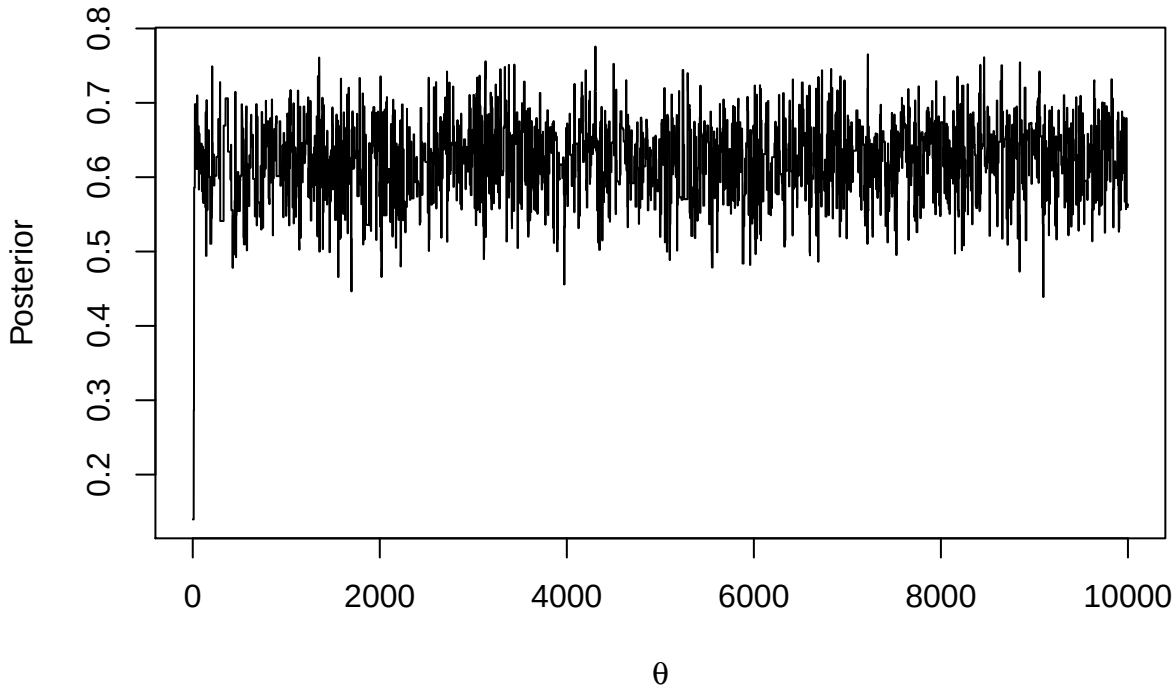


The histogram seems to agree with the target curve

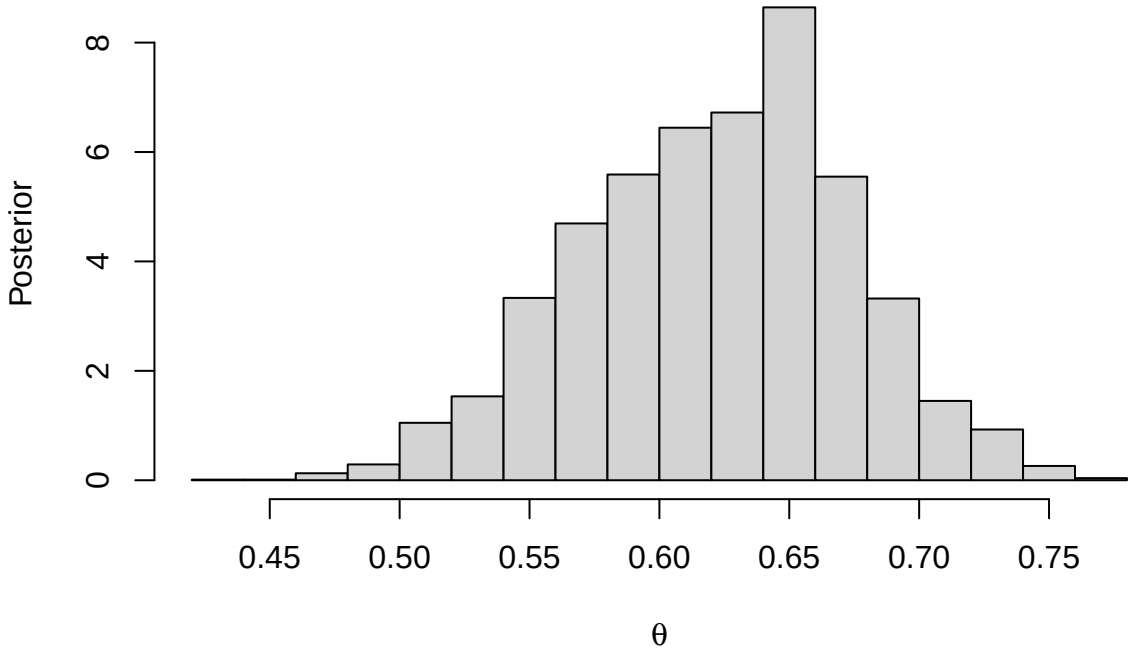
Question 3

```
f3 = function(the, x) {
  if(the<0 || the>1) {
    return(0)
  }
  return((the+2)^x[1]*(1-the)^(x[2]+x[3])*the^x[4])
}

xs = c(125,18,20,34)
m = 10000
k = 0
x = numeric(m)
x[1] = runif(1)
u = runif(m)

for(t in 2:m) {
  xt = x[t-1]
  alpha = xt/(-xt+1)
  y = rbeta(1,alpha,1)
  numerator = f3(y,xs)*dbeta(xt,y/(1-y),1)
  denominator = f3(xt,xs)*dbeta(y,alpha,1)

  if(numerator/denominator >= u[t]) {
    x[t] = y
```

```
  } else {
    x[t] = x[t-1]
    k = k+1
  }
}
plot(x, type='l',ylim = range(x),xlab=bquote(theta),ylab="Posterior")
```




```
bo1 = 1000
start = bo1 + 1
hist(x[start:m],prob=T,xlab = bquote(theta),ylab = 'Posterior',main = "")
```




```
theta_hat = mean(x[start:m])
theta_hat
```

```
## [1] 0.6219502
```

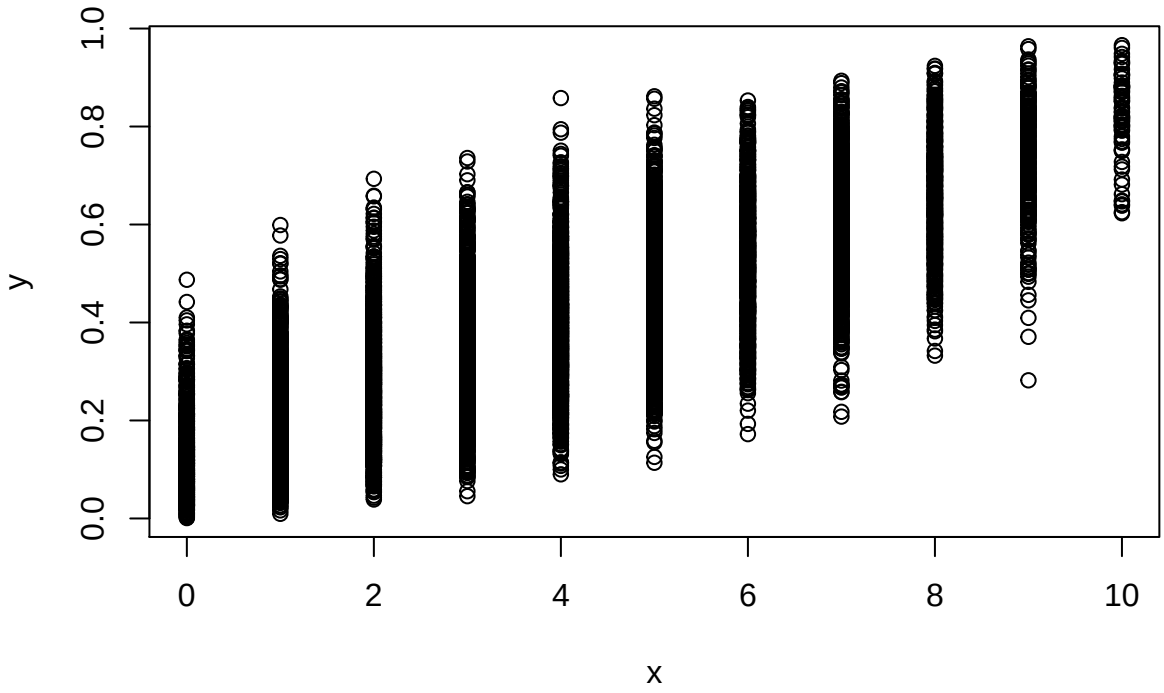
```
p_hat = c(theta_hat/4 + 0.5, (1-theta_hat)/4, (1-theta_hat)/4, theta_hat/4)
p_hat
```

```
## [1] 0.65548755 0.09451245 0.09451245 0.15548755
```

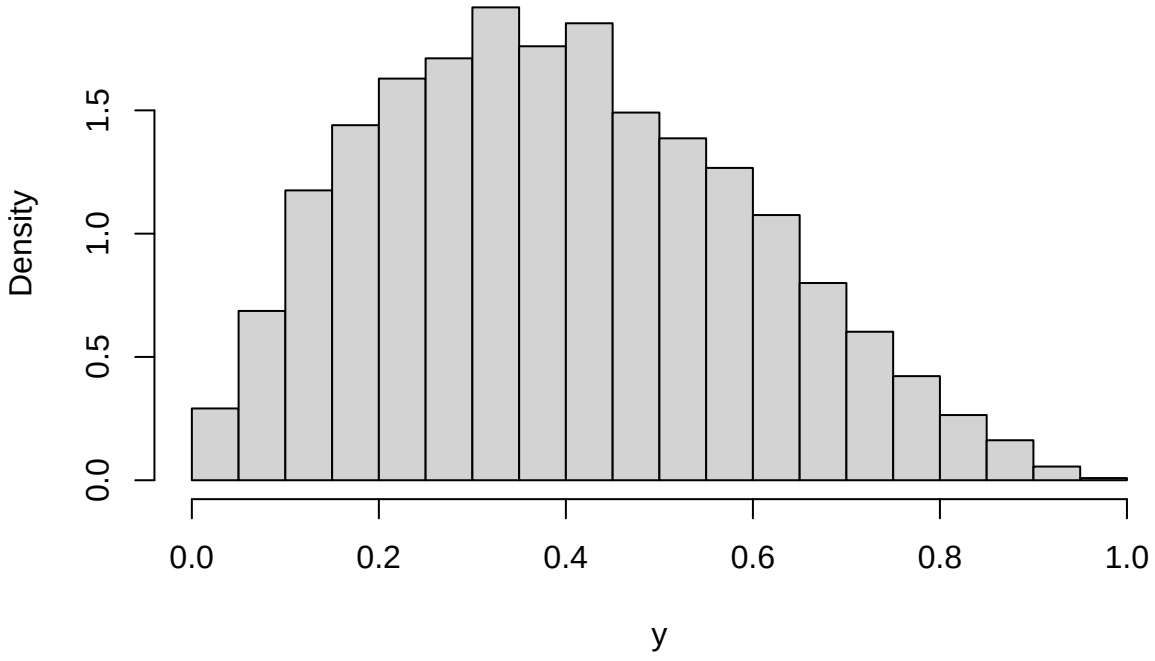
Question 4

```
f4 = function(m, a, b, n){
  X = matrix(0, nrow=m, ncol=2)
  y = (0.5*n + a)/(n + a + b)
  x = floor( n*y )
  X[1,] = c( x,y )
  for (t in 2:m) {
    y = X[t-1, 2]
    X[t, 1] = rbinom( 1, size=n, y)
    x = X[t, 1]
    X[t, 2] = rbeta( 1, x+a, n-x+b )
  }
  return( X )
```

```
}
m = 10000 #size of chain
b0 = 1000 #burn-in
a=2
b=3
n = 10
XYGibbs = f4(m, a, b, n)
aftburn = b0+1
plot( XYGibbs[aftburn:m,], xlab='x', ylab='y' )
```




```
hist( XYGibbs[aftburn:m,2], prob=T, main='', xlab='y' )
```




```
xGibbs = XYGibbs[aftburn:m,1]
fx_hat = table( xGibbs )/length( xGibbs )
round( fx_hat,3 )
```

```
## xGibbs
##     0     1     2     3     4     5     6     7     8     9    10
## 0.064 0.110 0.137 0.151 0.145 0.128 0.100 0.079 0.052 0.025 0.008
```

The plot is skewed at the right because it has a right tail

Question 5

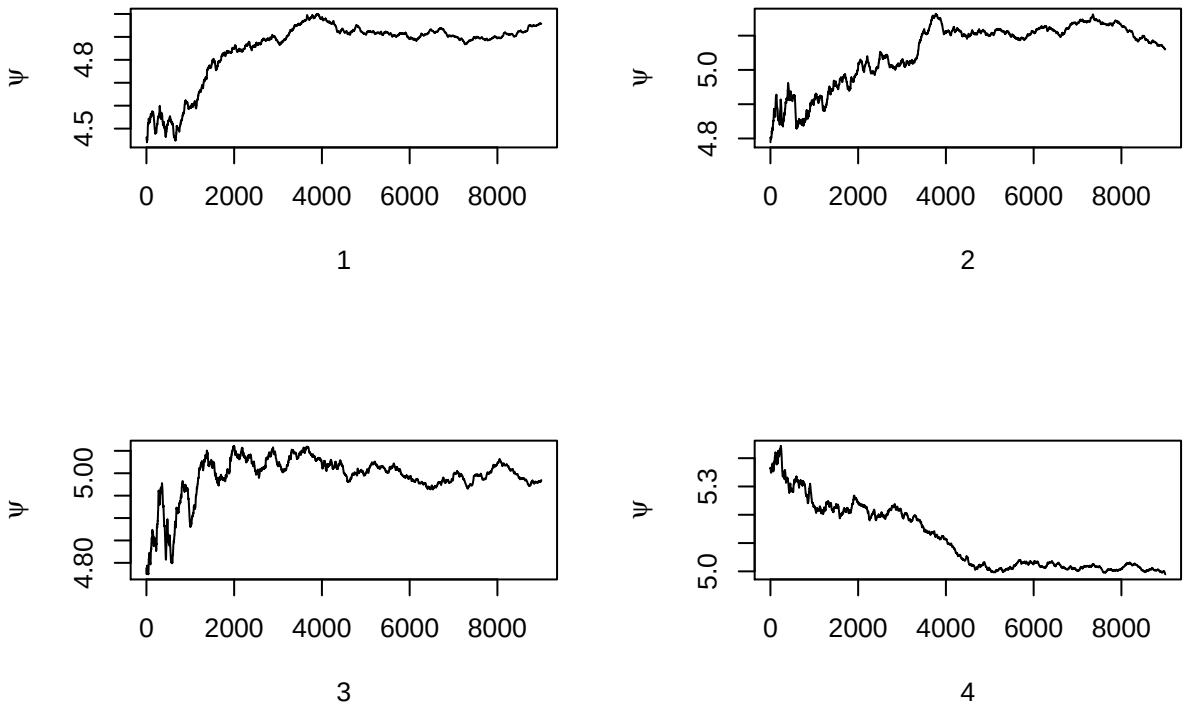
```
#declare all functions

Gelman.Rubin = function(psi) {
  psi = as.matrix(psi)
  n = ncol(psi)
  k = nrow(psi)
  psi.means = rowMeans(psi)
  B = n * var(psi.means)
  psi.w = apply(psi, 1, "var")
  W = mean(psi.w)
  v.hat = W*(n-1)/n + (B/n)
  r.hat = v.hat / W
  return(r.hat)
```

```
}

rayleigh.chain = function(sigma, M, X1) {
  X = rep(0, M)
  X[1] = X1
  for (i in 2:M) {
    xt = X[i-1]
    y = rchisq(1, df = xt)
    p = (dchisq(xt, y) * f(y)) / (dchisq(y, xt) * f(xt))
    if (runif(1) <= p) {X[i] = y} else {X[i] = xt}
    }
  return(X)
}
```

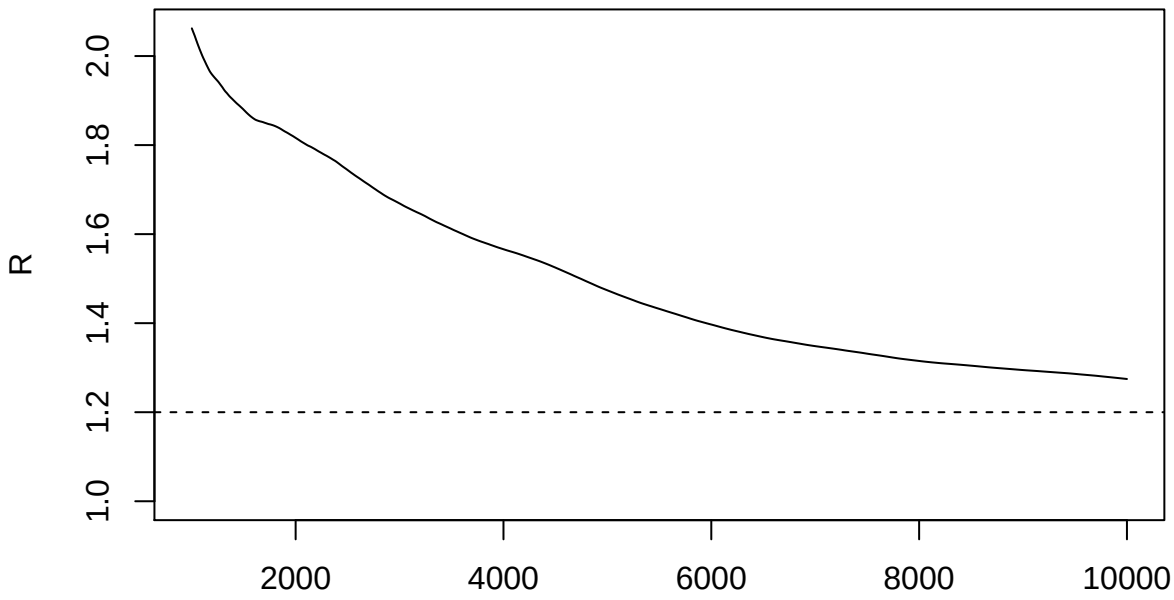
```
m = 10000
burn = 1001
index = burn:m
k = 4
X = matrix(0, nrow=k, ncol=m)
sigma = 4
x0 = c(1, 5, 10, 20)
# SAMPLING
for (i in 1:k) X[i,] = rayleigh.chain(sigma, m, x0[i])

# RESULTS
psi = t(apply(X, 1, cumsum))
for (i in 1:nrow(psi)) psi[i,] = psi[i,]/(1:ncol(psi))
print(Gelman.Rubin(psi))
```

```
## [1] 1.274579
```

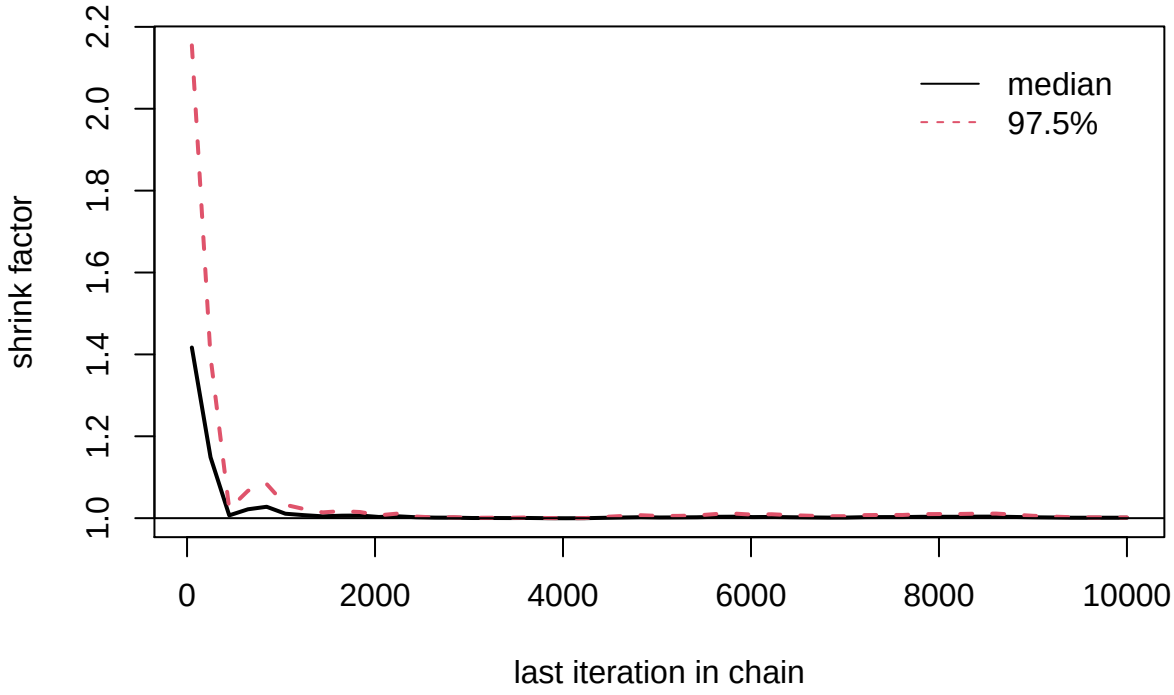
```
par(mfrow=c(2,2))
for (i in 1:k) plot(psi[i, index], type='l', xlab=i, ylab=bquote(psi))
```




```
par(mfrow=c(1,1))
rhat = rep(0, m)
for (j in index) rhat[j] = Gelman.Rubin(psi[,1:j])
plot(index, rhat[index], type='l', xlab="", ylab="R", ylim=c(1, max(rhat)))
abline(h=1.2, lty=2)
```




```
#install.packages("coda")
library(coda)
X1 = as.mcmc(X[1,])
X2 = as.mcmc(X[2,])
X3 = as.mcmc(X[3,])
X4 = as.mcmc(X[4,])
Xlst = mcmc.list(X1, X2, X3, X4)
gelman.plot(Xlst, col = c(1,2), lwd = c(2, 2))
```


shrink factor
1.0
1.2
1.4
1.6
1.8
2.0
2.2
median
97.5%
0
2000
4000
6000
8000
10000
last iteration in chain