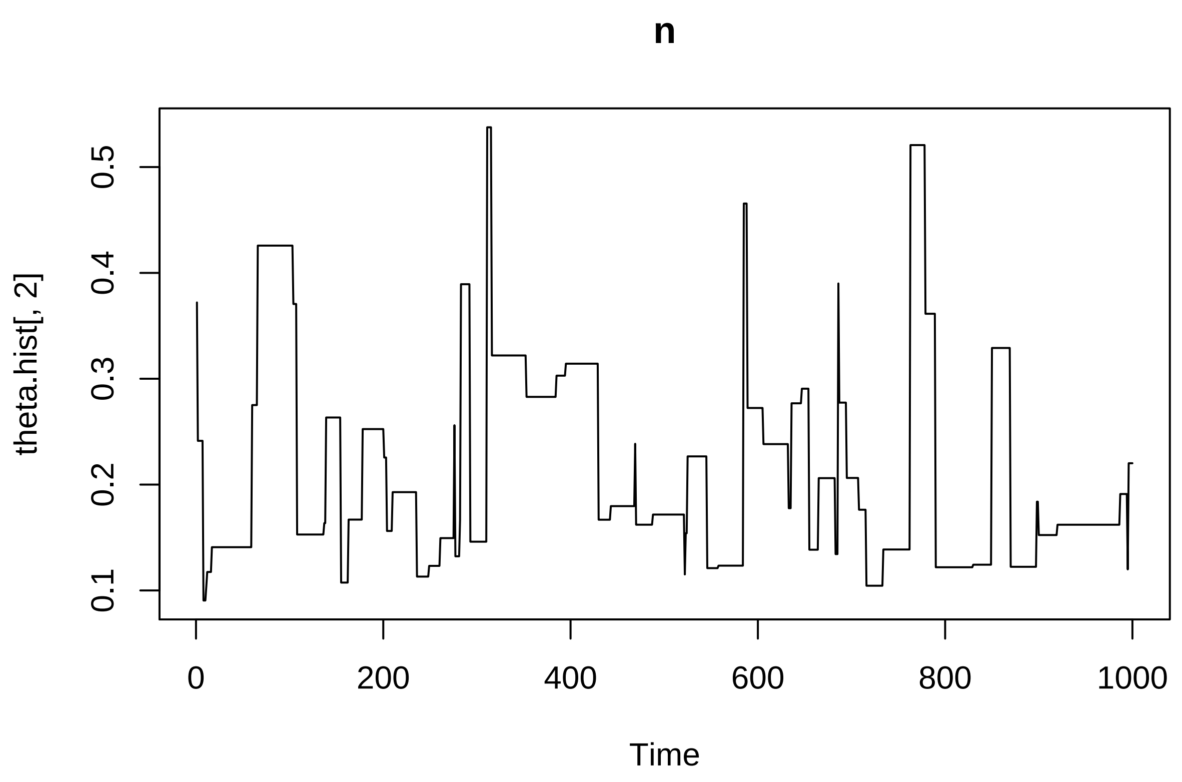
a2q3

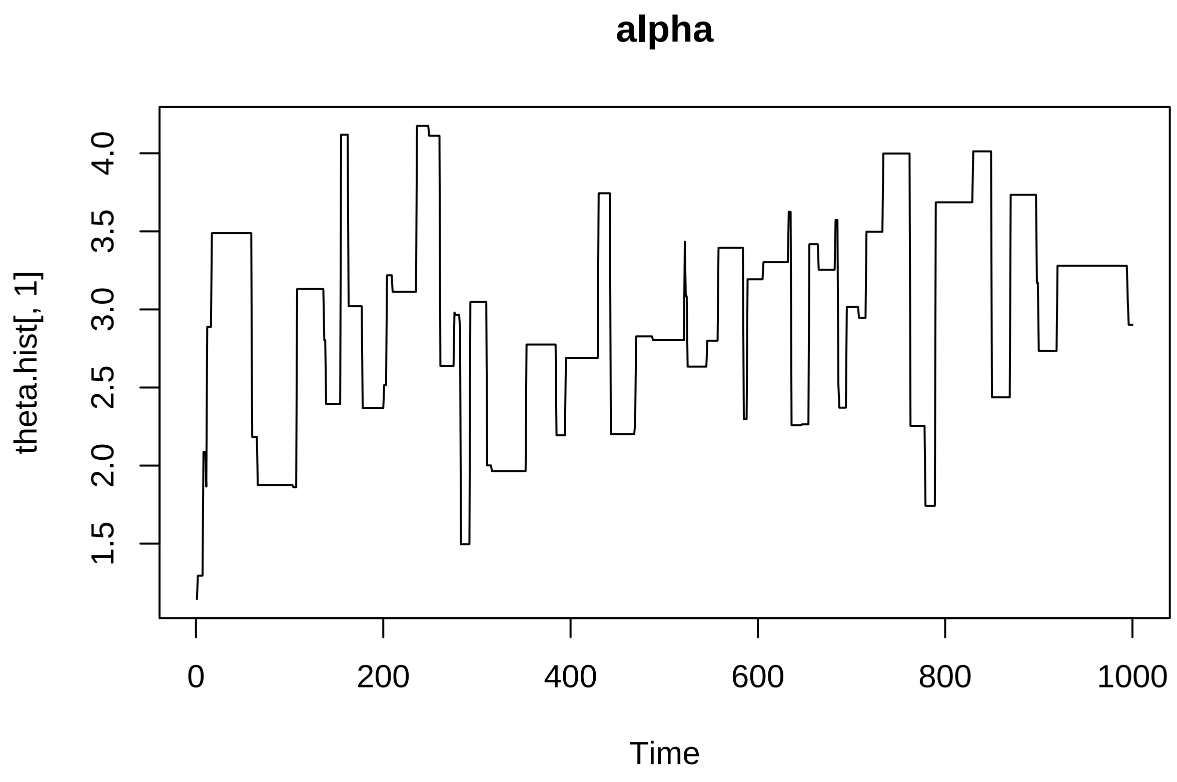
```
M=1000

# acceptance rate:
x=c(0.56, 2.26, 1.90, 0.94, 1.40, 1.39, 1.00, 1.45, 2.32, 2.08, 0.89, 1.68)
x_prod = prod(x)
x_sum = sum(x)
alpha = function(alpha_prev, n_prev, alpha_aft, n_aft, c, beta){
  num1 = alpha_aft^12*n_aft^(beta+11)*(x_prod)^(alpha_aft-1)*exp(sum(-n_aft*x^alpha_aft) -alpha_aft-c*n_
  den1 = alpha_prev^12*n_prev^(beta+11)*(x_prod)^(alpha_prev-1)*exp(sum(-n_prev*x^alpha_prev) -alpha_pre
  num2 = 1/(alpha_aft*n_aft)*exp(-alpha_prev/alpha_aft-n_prev/n_aft)
  den2 = 1/(alpha_prev*n_prev)*exp(-alpha_aft/alpha_prev-n_aft/n_prev)
  num1/den1*num2/den2
}

markov.next = function(theta){
  alpha.aft= rexp(1,1/theta[1])
  n.aft = rexp(1,1/theta[2])
  c(alpha.aft, n.aft)
}


# initialize some value for c and beta
beta=1
c=10

theta = c(1,1)
theta.hist = matrix(0, nrow = M, ncol = 2)
for (i in 1:M){
  new = markov.next(theta)
  acc.rate = alpha(theta[1], theta[2], new[1], new[2], c, beta)

  r = runif(1)
  if (acc.rate>r){
    theta.hist[i,]=new
    theta = new
  }else{
    theta.hist[i,]=theta
  }

}
theta.hist = na.omit(theta.hist)

n=nrow(theta.hist)
n
```

```
## [1] 1000
```

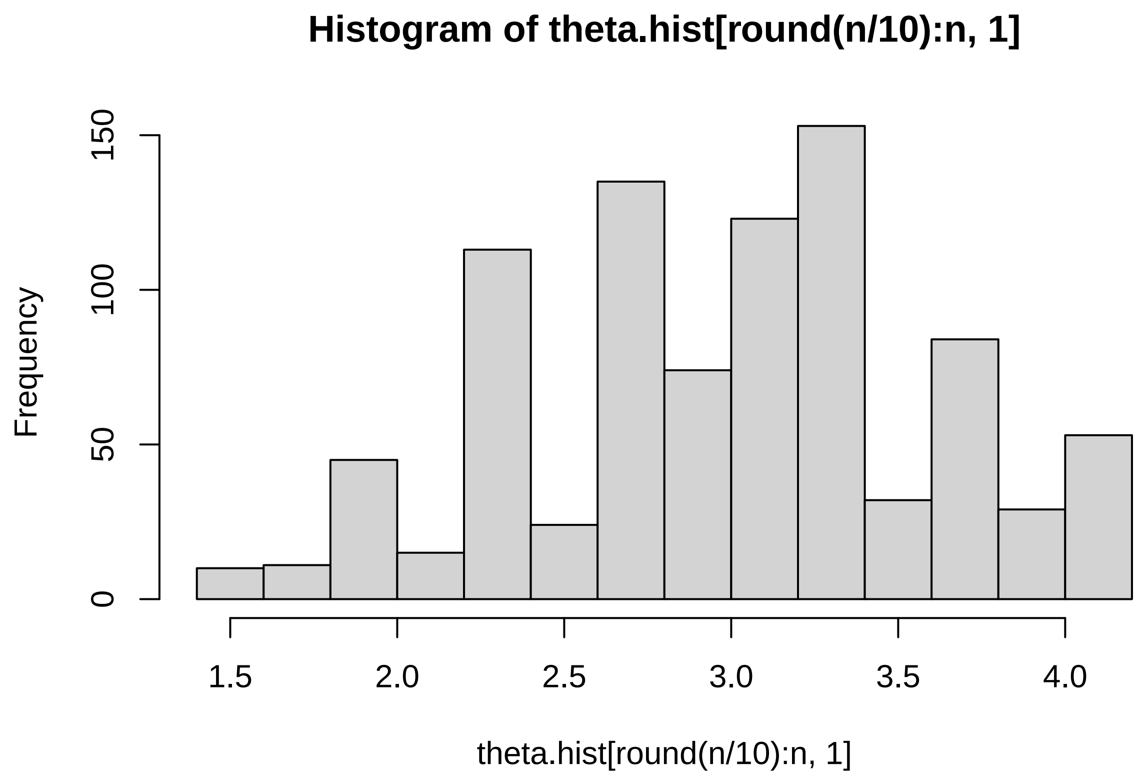
```r
ts.plot(theta.hist[,2], main = 'n')
```



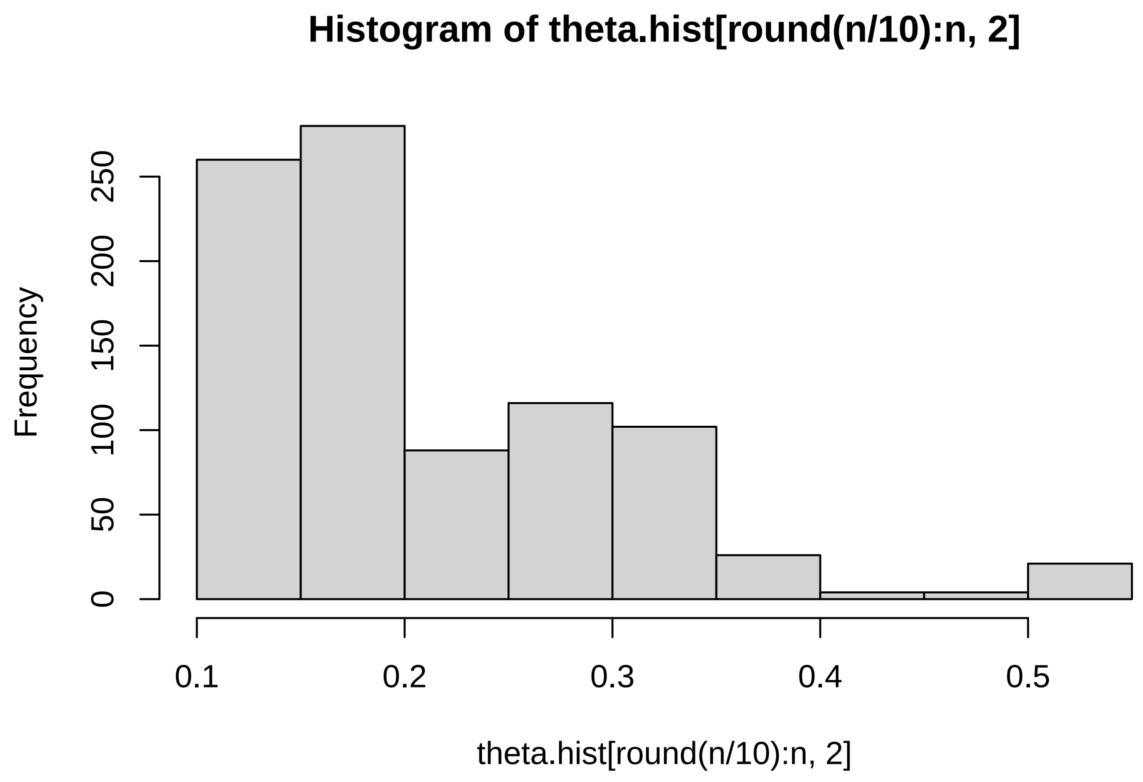
```r
ts.plot(theta.hist[,1], main = 'alpha')
```



```r
hist(theta.hist[round(n/10):n,1])
```



```
hist(theta.hist[round(n/10):n,2])
```



```
# we just use the last 9/10 of approximate, as the first are depending on the initial values of theta
alpha.hat = mean(theta.hist[round(n/10):n,1])
n.hat = mean(theta.hist[round(n/10):n,2])

alpha.hat
```

```
## [1] 2.985396
```

```
n.hat
```

```
## [1] 0.2085094
```

```
#95% confidence interval:
#sigma hat
alpha.se = sd(theta.hist[round(n/10):n,1])/sqrt(9/10*n)
n.se = sd(theta.hist[round(n/10):n,2])/sqrt(9/10*n)
alpha.se
```

```
## [1] 0.02037896
```

```
n.se
```

```
## [1] 0.003009696
```

```
alpha.ci = alpha.hat + c(1,-1)*alpha.se*1.96
n.ci = n.hat + c(1,-1)*n.se*1.96
alpha.ci
```

```
## [1] 3.025339 2.945453
```

```
n.ci
```

```
## [1] 0.2144084 0.2026104
```

```
getmode <- function(v) {
   uniqv <- unique(v)
   uniqv[which.max(tabulate(match(v, uniqv)))]
}


getmode(round(theta.hist[round(n/10):n,1],1 ) )
```

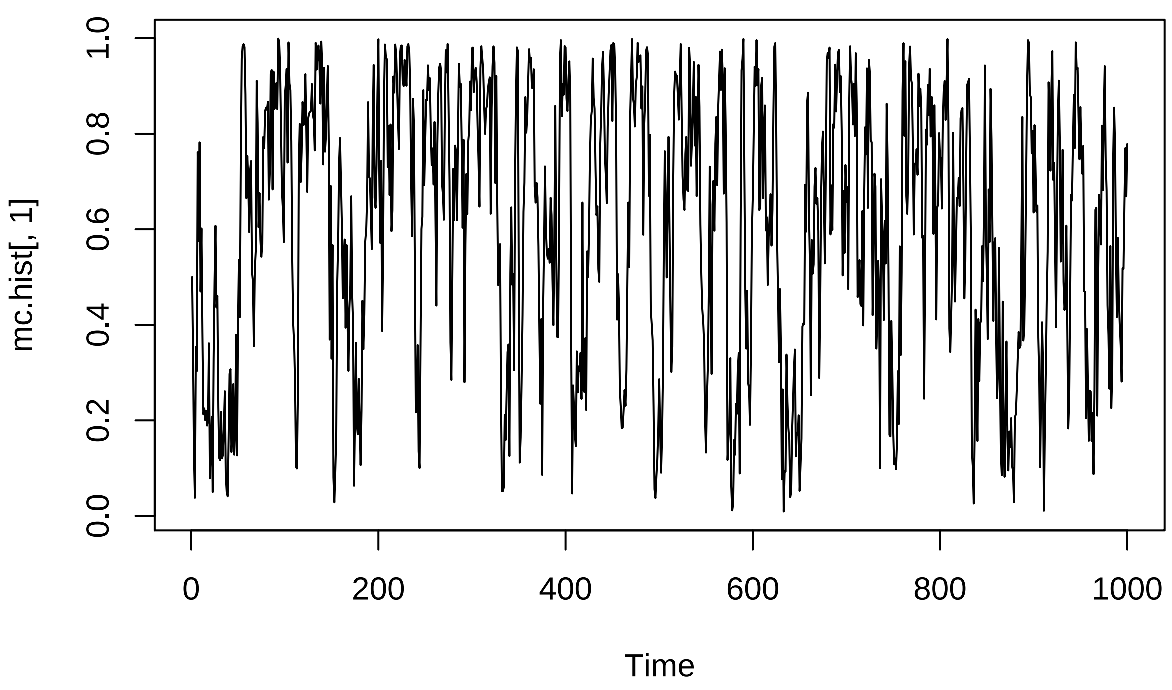
```
## [1] 3.3
```

```
getmode(round(theta.hist[round(n/10):n,2],1))
```

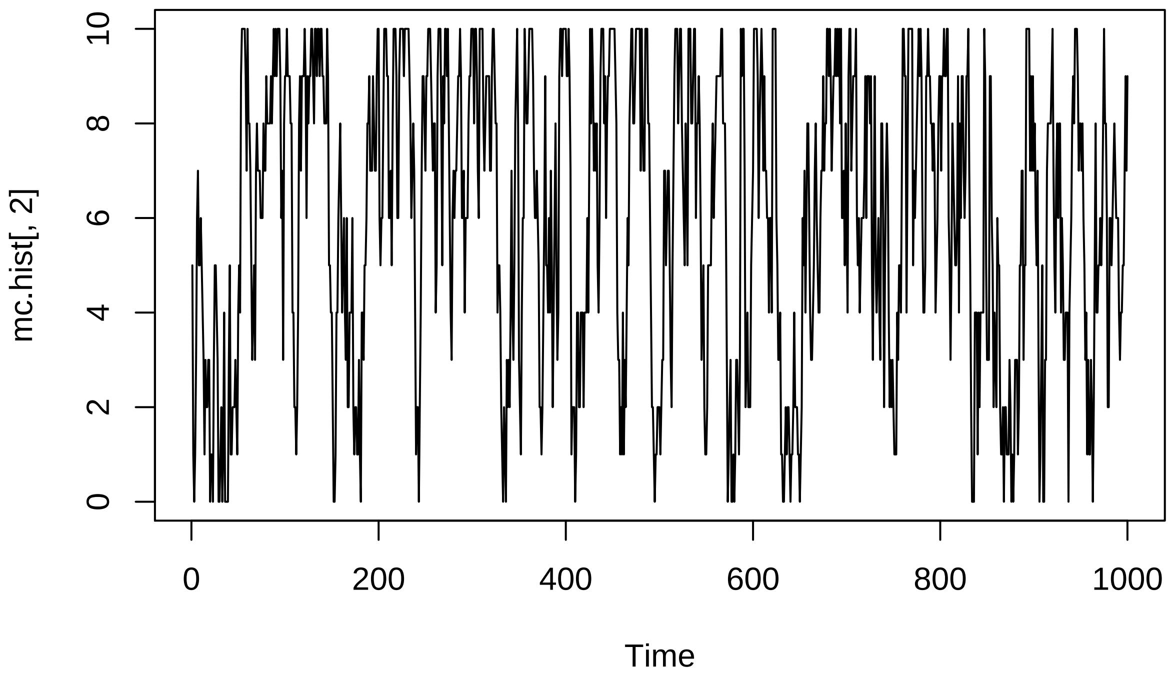
```
## [1] 0.2
```

4(a)

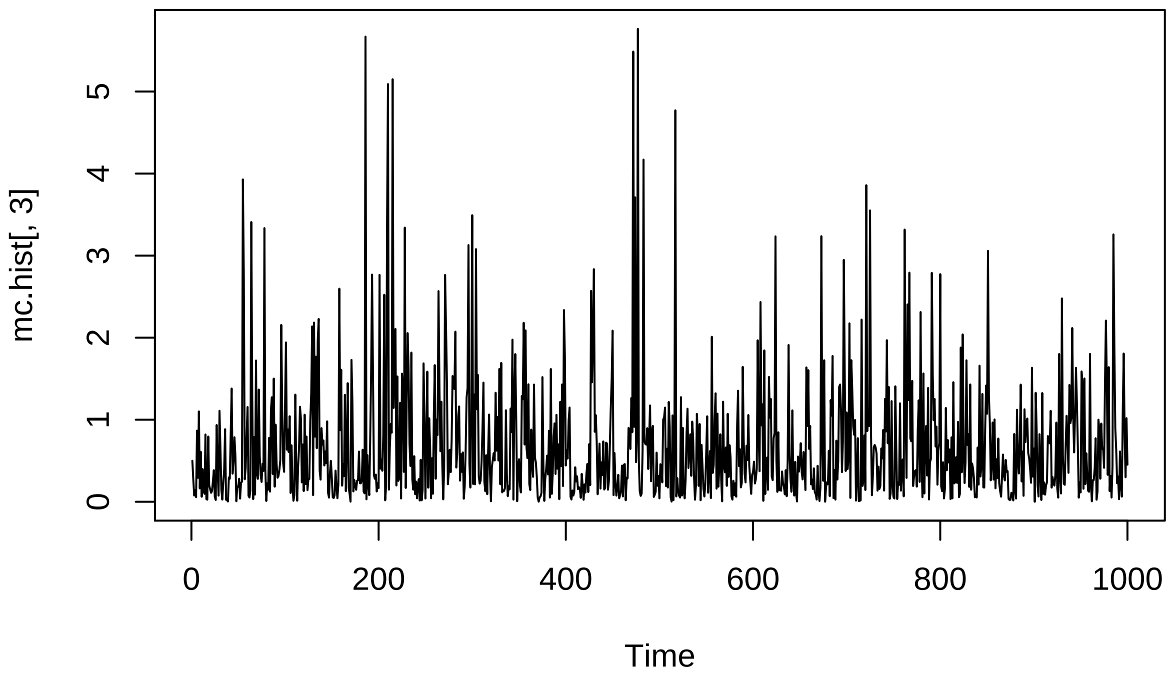
```
# generate x from y z

M=1000

getx = function(x,y,z){
  rbeta(1,y+z+1, 11-y)
}

gety = function(x,y,z){
  rbinom(1,10,x)
}




getz = function(x,y,z){
  for (i in 1:M){
    newz = rexp(1, 1/z)
    alpha = (exp(-1)*x)^(newz-z)  / dexp(newz, 1/z)* dexp(z, 1/newz)
    if (runif(1) < alpha){
      z = newz
    }
  }
  z
}


mc.hist = matrix(0, nrow = M, ncol = 3)
mc.hist[1,]= c(0.5,5,0.5)
for (i in 1:(M-1)){
  mc.hist[i+1,1] = getx(mc.hist[i,1], mc.hist[i,2], mc.hist[i,3])
  mc.hist[i+1,2] = gety(mc.hist[i+1,1], mc.hist[i,2], mc.hist[i,3])
  mc.hist[i+1,3] = getz(mc.hist[i+1,1], mc.hist[i+1,2], mc.hist[i,3])
}

n = nrow(mc.hist)

ts.plot(mc.hist[,1])
```



```r
ts.plot(mc.hist[,2])
```



```r
ts.plot(mc.hist[,3])
```



```r
#d  with our chain, just calculate z^2*y^3*exp(-x^2) for each iteration, then get the average

x=mc.hist[400:n,1]
y=mc.hist[400:n,2]
z=mc.hist[400:n,3]
mean(z^2*y^3*exp(-x^2))
```

```
## [1] 248.598
```

it seems the chain converges around 400th iteration