Final Homework

2019711351 Choi Teayoung

1 From Efron's Paper

1.1 Verify that (4) is the same as (2).

Proof.

$$\begin{aligned}
\hat{\sigma}_J &= [\frac{n-1}{n}\sum_{i=1}^{n}(\bar{x}_{(i)} - \bar{x}_{(\cdot)})^2]^{1/2} \cdots (4) \\
&= [\frac{n-1}{n}\sum_{i=1}^{n}(\frac{n\bar{x} - x_i}{n-1} - \bar{x})^2]^{1/2} \\
&(\because \bar{x}_{(i)} = \frac{n\bar{x} - x_i}{n-1}) \\
&(\because \bar{x}_{(\cdot)} = \frac{\sum_1^n x_{(i)}}{n} = \frac{\sum_1^n (n\bar{x} - x_i)}{n} = \bar{x}) \\
&= [\frac{n-1}{n}\sum_{i=1}^{n}(\frac{n\bar{x} - x_i - (n-1)\bar{x}}{n-1})^2]^{1/2} \\
&= [\frac{n-1}{n}\sum_{i=1}^{n}(\frac{\bar{x} - x_i}{n-1})^2]^{1/2} \\
&= [\frac{1}{n(n-1)}\sum_{i=1}^{n}(x_i - \bar{x})^2]^{1/2} \cdots (2)
\end{aligned}$$

□

1.2 Show that (9) equals (8) times $n^2/(n^2-1)$

Proof.

$$\begin{aligned}
(9) \cdots \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x}_{(i)})^2 &= \frac{1}{n}\sum_{i=1}^{n}(x_i - \frac{n\bar{x} - x_i}{n-1})^2 \\
&= \frac{1}{n}\sum_{i=1}^{n}(\frac{(n-1)x_i}{n-1} - \frac{n\bar{x} - x_i}{n-1})^2 \\
&= \frac{1}{n}\sum_{i=1}^{n}(\frac{n(x_i - \bar{x})}{n-1})^2 \\
&= \frac{n}{(n-1)^2}\sum_{i=1}^{n}(x_i - \bar{x})^2 \cdots (A) \\
&(\because \hat{\sigma}^2 = \frac{1}{n(n-1)}\sum_{1}^{n}(x_i - \bar{x})^2) \\
&= (n+1) \cdot (\frac{n-1}{n^2} \cdot A) \cdot \frac{n^2}{n-1} \cdot \frac{1}{n+1} \\
&= (n+1)\hat{\sigma}^2 \cdot \frac{n^2}{n^2-1} \cdots (8)
\end{aligned}$$

□

1.3 Show that $var.\hat{\theta_L}(P^*) = var.(P^* - P^0)'U = \sum U_i^2/n^2$ using the covariance matrix for (12) and the fact that $\sum U_i = 0$

$$P^* \ \frac{1}{n} MVN_n(n, P_0)$$

with $P_0 = \frac{1}{n}(1, \cdots 1)^T$

$$\begin{aligned}
\hat{\theta_L}(P^*) &= \hat{\theta}_{(.)} + (P^* - P^0)'\text{U이고}, \\
E(P^*) &= nP_i, \\
var(P^*) &= \frac{P_2(1-P_1)}{n} = \frac{n-1}{n^2}, \\
Cov(P_i^*, P_i^*) &= -\frac{P_i P_i}{n} = -\frac{1}{n^3}
\end{aligned}$$

따라서

Proof.

$$\begin{aligned}
Var(\hat{\theta}_L(P^*)) &= Var(\hat{\theta}_{(.)} + (P^* - P^0)'U) = Var(P^{*\prime}U) \\
&= \frac{1}{n^3}((n-1)U_1 - U_2 \cdots - U_n)(U_1 \cdots U_n)' \\
&= \frac{1}{n^3}(nU_1 \cdots nU_n)(U_1 \cdots U_n)' \\
&= \frac{1}{n^3}(nU_1^2 + \cdots + nU_n^2) \\
&= \frac{1}{n^2}\sum U_i^2 \\
\therefore var.\hat{\theta_L}(P^*) &= var.(P^* - P^0)'U = \sum U_i^2/n^2
\end{aligned}$$

□

1.4 If we set $\hat{\theta_L}(P) = \hat{\theta_{(.)}}(P - P^0)'U$ as in (14), then show that $\hat{\theta_L}(P_{(1)}) = \theta(\hat{P}_{(1)}) = \hat{\theta_{(1)}}$

Proof.

$$\begin{aligned}
\hat{\theta_L}(P) &= \hat{\theta_{(.)}}(P - P^0)'U \\
&= \hat{\theta_{(.)}} + (\frac{1}{n} \cdots \frac{1}{n})(\hat{\theta_{(.)}} - \hat{\theta_{(j)}} \cdots \hat{\theta_{(.)}} - \hat{\theta_{(j)}})' \\
&= \hat{\theta_{(.)}} + \hat{\theta_{(.)}} - \frac{1}{n}\sum \hat{\theta_{(j)}} \\
\hat{\theta_{(.)}} &= \frac{1}{n}\sum \hat{\theta_{(j)}} = \frac{1}{n}\sum \theta_{(\hat{P}_{(i)})} = \frac{2}{n}\sum \hat{\theta_{(i)}} - \frac{1}{n}\sum \hat{\theta_{(i)}} = \frac{1}{n}\sum \hat{\theta_{(j)}} = \frac{1}{n}\sum \hat{\theta_{P_{(i)}}}
\end{aligned}$$

$\therefore \hat{\theta_L}(P_{(1)}) = \theta(\hat{P}_{(1)}) = \hat{\theta_{(1)}}$

□

2 문제 11.5

Write a function to solve the equation

$$\frac{2\Gamma(\frac{k}{2})}{\sqrt{\pi(k-1)}\Gamma(\frac{k-1}{2})}\int_0^{c_{k-1}}\left(1+\frac{u^2}{k-1}\right)^{-k/2}du$$
$$=\frac{2\Gamma(\frac{k+1}{2})}{\sqrt{\pi k}\Gamma(\frac{k}{2})}\int_0^{c_k}\left(1+\frac{u^2}{k}\right)^{-(k+1)/2}du$$

for a, where

$$c_k=\sqrt{\frac{a^2k}{k+1-a^2}}.$$

Compare the solutions with the points A(k) in Exercise 11.4

2.1 접근법

식의 적분 부분을 수치적분 중 직사각형 법을 이용하여 적분을 실시하고 뉴튼 랩슨 방법을 이용하여 a에 대한 해를 구한다. 그리고 Exercise 11.4의 결과와 비교한다.

```
bi1 = function(k, a, b, epsilon) {
if ( sqrt(k-1)*gamma((k-1)/2)<=0 || sqrt(k)*(gamma(k/2))<=0 ) print("wrong initial values")
if ( k+1-a^2 <= 0) print("wrong initial values")


f = function(k) {
( 2 * exp(lgamma(k/2)) ) / (sqrt( pi * (k-1)) * exp(lgamma( (k-1)/2 )))
}
c_k = function(k, a) { sqrt( (a^2 * (k-1) / (k-a^2) ))}


tae0=function(c,d,k)
{ n<-10000
h<-(d-c)/n
integral<-(z(c,k)+z(d,k))/2
x<-c
for(i in 1:(n-1)){
x<-x+h
integral<-integral+z(x,k)
}
integral<-integral*h
return(integral)
}

z=function(u,k) {(1 + u^2/(k-1))^(-(k)/2)}
i<-function(k,a){tae0(0,c_k(k,a),k)}
h = function(k, a) {
```

```
f(k) * i(k,a) - f(k + 1) * i(k+1,a)
}

hx0<-h(k,a)
hx1<-h(k,b)
if(hx0*hx1>0) return("wrong initial values")
e=abs(b-a)
N = 1 # # of iteration for solution

while(e>epsilon) {
  N = N + 1
  e=e/2
  c<-(a+b)/2
  hx2=h(k,c)
  if(hx0*hx2<0){
  b=c ; hx1=hx2
  }
  else{
  a=c ; hx0=hx2
  }
  print(c)
 }
}
bi1(100,0,3, 0.000001)
```

```
## [1] 1.5
## [1] 2.25
## [1] 1.875
## [1] 1.6875
## [1] 1.78125
## [1] 1.734375
## [1] 1.710938
## [1] 1.722656
## [1] 1.716797
## [1] 1.719727
## [1] 1.721191
## [1] 1.720459
## [1] 1.720825
## [1] 1.720642
## [1] 1.720551
## [1] 1.720596
## [1] 1.720619
## [1] 1.720608
## [1] 1.720602
## [1] 1.720599
## [1] 1.720598
## [1] 1.720598
```

```
### ex11.4
cpn <- function(k,a,epsilon,n){
if(k-a^2<=0 || k+1-a^2<=0) print("wrong initial values")
Ck <- function(a,k){
sqrt(a^2*(k-1)/(k-a^2))
}
```

```
Sk <-function(a,k){
1-pt(Ck(a,k),k-1)
}
eqf2<- function(a,k){
Sk(a,k)-Sk(a,k+1)
}
e=1
N=1
while(e>epsilon){
N<-N+1
if(N>n) return("not converge")
d <- epsilon
a1 <- a-eqf2(a,k)*d/(eqf2(a+d,k)-eqf2(a,k))
e <- abs(a1-a)
a=a1 ## a  a1
if (k+1-a^2 <= 0) break
}
return(a1)
}

cpn(25,1,0.00001,100000)
```

```
## [1] 1.687347
```

```
cpn(100,1,0.00001,100000)
```

```
## [1] 1.720599
```

```
cpn(500,1,0.00001,100000)
```

```
## [1] 1.729745
```

```
cpn(1000,1,0.00001,100000)
```

```
## [1] 1.730897
```

2.2 결론

연습문제 11.5의 결과인 “1.730897”과 11.4의 결과인 “1.720599”가 모두 비슷하다. k의 값이 더 커지면 결과도 거의 비슷하게 수렴한다는 사실을 알 수 있다.

3 문제 11.6

Write a function to compute the cdf of the Cauchy distribution, which has density

$$\frac{1}{\theta\pi(1+[(x-\eta)/\theta]^2)}, \qquad -\infty < x < \infty,$$

where $\theta > 0$. Compare your results to the results from the R function `pcauchy`. (Also see the source code in `pcauchy.c`.)

3.1 접근법

수치 적분 방법 중 사다리꼴방법을 이용하여 적분을 하고, 그 적분 결과값과 실제로 pcauchy함수를 이용하여 구한 값이 차이가 있는지 확인한다. 사다리꼴방법은 적분하고자 하는 함수의 형태에 직사각형 보다 조금 더 비슷한 사다리꼴을 사용하는 방법이다.

```
f<-function(theta, x, eta) {1/(theta*pi*(1+((x-eta)/theta)^2))} ##cdf

tae0=function(theta,a,b,eta,n)
{
h<-(b-a)/n
integral<-(f(theta,a,eta)+f(theta,b,eta))/2
x<-a
for(i in 1:(n-1)){
x<-x+h
integral<-integral+f(theta,x,eta)
}
integral<-integral*h
return(integral)
}
tae0(1,-2,2,2,50) #n=50
```

```
## [1] 0.4220162
```

```
tae0(1,-2,2,2,100) #n=100
```

```
## [1] 0.4220197
```

```
tae0(1,-2,2,2,1000) #n=1000
```

```
## [1] 0.4220209
```

```
pcauchy(2,2,1)-pcauchy(-2,2,1) #     interval  f(-inf,2)-f(-2,inf)
```

```
## [1] 0.4220209
```

3.2 결론

사다리꼴 방법을 이용하여 적분한 값과 실제값이 비슷한 것을 알 수 있다. 또한, 구간의 개수가 점점 늘어날수록 값이 더 비슷해지는 것을 알 수 있다. pcauchy 함수의 interval을 이용해 적분값을 계산

4 문제 9.3

Use the Metropolis-Hastings sampler to generate random variables from a standard Cauchy distribution. Discard the first 1000 of the chain, and compare the deciles of the generated observations with the deciles of the standard Cauchy distribution (see qcauchy or qt with df=1). Recall that a Cauchy (θ, η) distribution has density function

4.1 접근법

교재249쪽부터 나와 있는 Metropolis-Hastings sampler 방법을 이용한다. 메트로폴리스 알고리즘은 대칭인 전치행렬을 임의로 하나 선정하고, 가역성을 만족시키기 위해 적당한 X_0를 생성한다.그리고 체인이 어떤 기준에 따라 고정 된 분포로 수렴 할 때까지 다음 알고리즘을 반복한다.

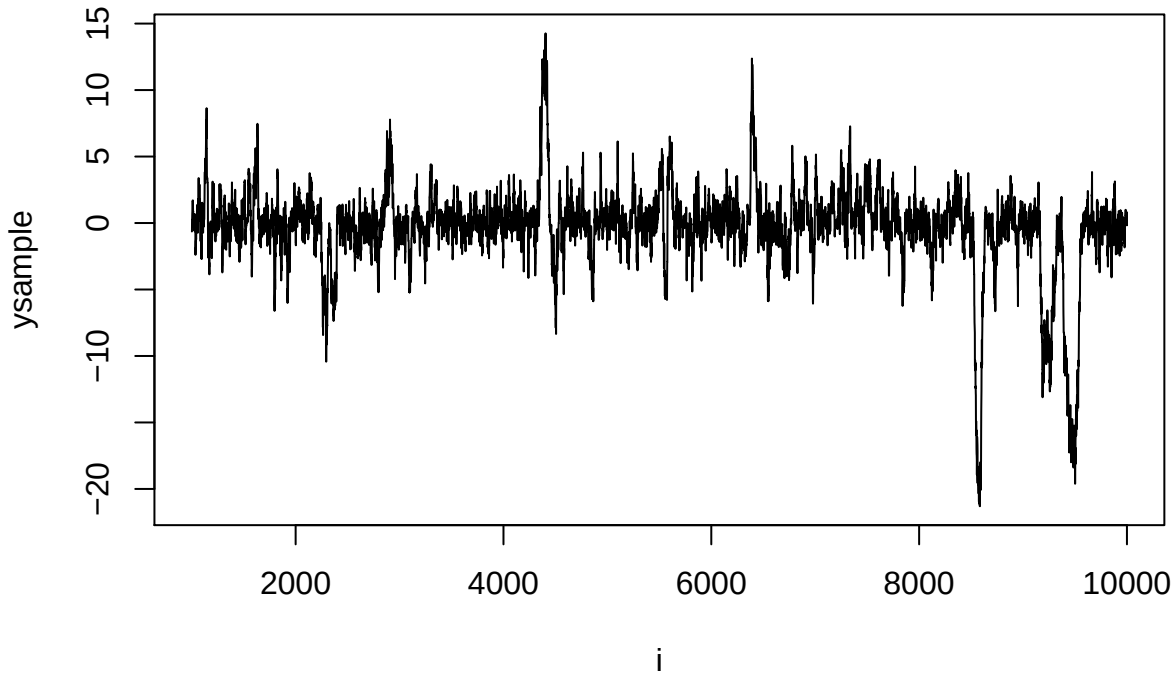
$1, g(\cdot|X_t)$으로부터 Y생성

$2, U(0,1)$로부터 기준 난수 생성

$3, If U \leq \frac{f(Y)g(X_t|Y)}{f(X_t)g(Y|X_t)}, then X_{t+1} = Y$

$4, else X_{t+1} = X_t$

```
  f <- function(x){1/(pi*(1+x^2))
  }
  m <- 10000
  x <- numeric(m)
  x[1] <- rnorm(1)
  k <- 0

  for (i in 2:m) {
      xt <- x[i-1]
      y <- rnorm(1,xt,1)
      num <- f(y) * dnorm(xt,y,1)
      den <- f(xt) * dnorm(y,xt,1)
      u <- runif(1) #test
      if (u <= num/den) x[i] <- y else{
        x[i] <- xt
        k <- k+1      #y is rejected
      }
      }

i <- 1001:10000

ysample <- x[i]
plot(i, ysample, type="l")
```




```
  realc<-rcauchy(9000)
  quantile(ysample, probs=seq(0.1, 1, length=10)) #MCMC  Deciles
```

```
##          10%          20%          30%          40%          50%
## -2.948802791 -1.393755548 -0.720234848 -0.334623790 -0.006451455
##          60%          70%          80%          90%         100%
##  0.303737516  0.656955183  1.195267193  2.314312753 14.265728943
```

```
  quantile(realc, probs=seq(0.1, 1, length=10)) #  Cauchy Deciles
```

```
##           10%           20%           30%           40%           50%
##   -2.93034705   -1.33318188   -0.71536949   -0.31282731    0.01308983
##           60%           70%           80%           90%          100%
##    0.32576685    0.72390984    1.37272827    2.97196450 3936.20479480
```

4.2 plot 설명

sample plot: proposal분포($X^2_{(1)}$)에서 10000를 추출한 뒤 1000개를 버리고 나서, cauchy분포를 따를 거라고 기대되는 9000개에 대한 sample plot.

4.3 결론

M-H알고리즘은 우리의 타겟 함수인 Cauchy(0,1)을 커버하지 못한다.

5 문제 9.8

9.8 This example appears in [40]. Consider the bivariate density

$$f(x,y) \propto \binom{n}{x} y^{x+a-1}(1-y)^{n-x+b-1}, \quad x = 0,1,\ldots,n,\ 0 \le y \le 1.$$

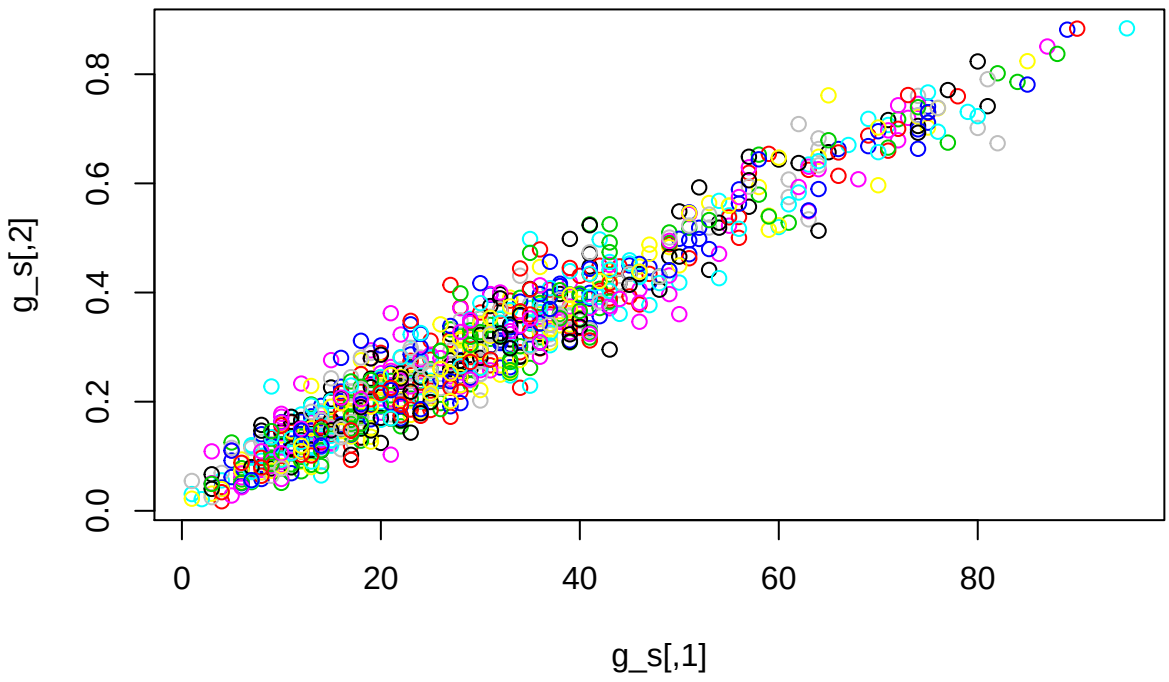
It can be shown (see e.g. [23]) that for fixed a, b, n, the conditional distributions are Binomial(n, y) and Beta$(x + a, n - x + b)$. Use the Gibbs sampler to generate a chain with target joint density $f(x, y)$.

5.1 접근법

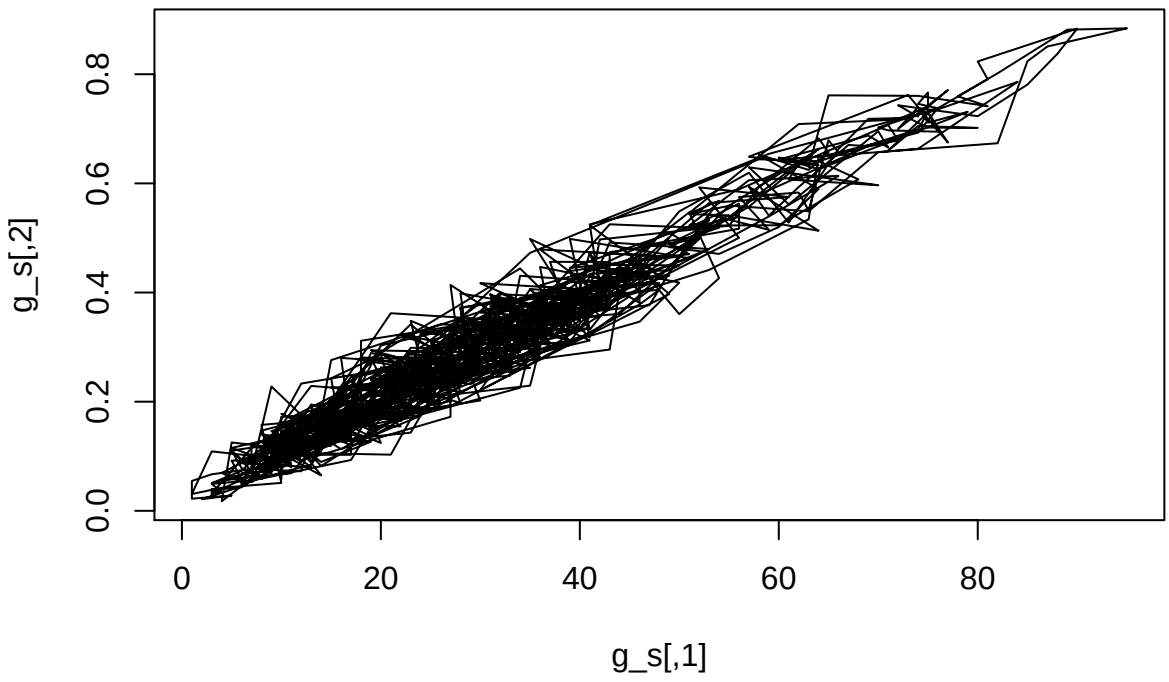
MCMC 방법 중 Gibbs sampler 방법을 이용한다. 이 방법은 주어진 밀도함수를 따르는 분포에서 직접 표본을 뽑지 못하고, 각각의 변수들에 대하여 다른 두 변수들이 주어졌을 때의 조건부 분포가 알려져 있으면 이를 이용하여 표본추출을 하는 방법이다.

알고리즘은 1. x_0, y_0의 초기값을 지정하고, 2. i번째 난수 벡터(x_i, y_i)가 주어졌을 때, $i+1$번째 난수 벡터를 조건부 분포에서 추출한다. 3. 이 과정을 N번 반복하면 된다.

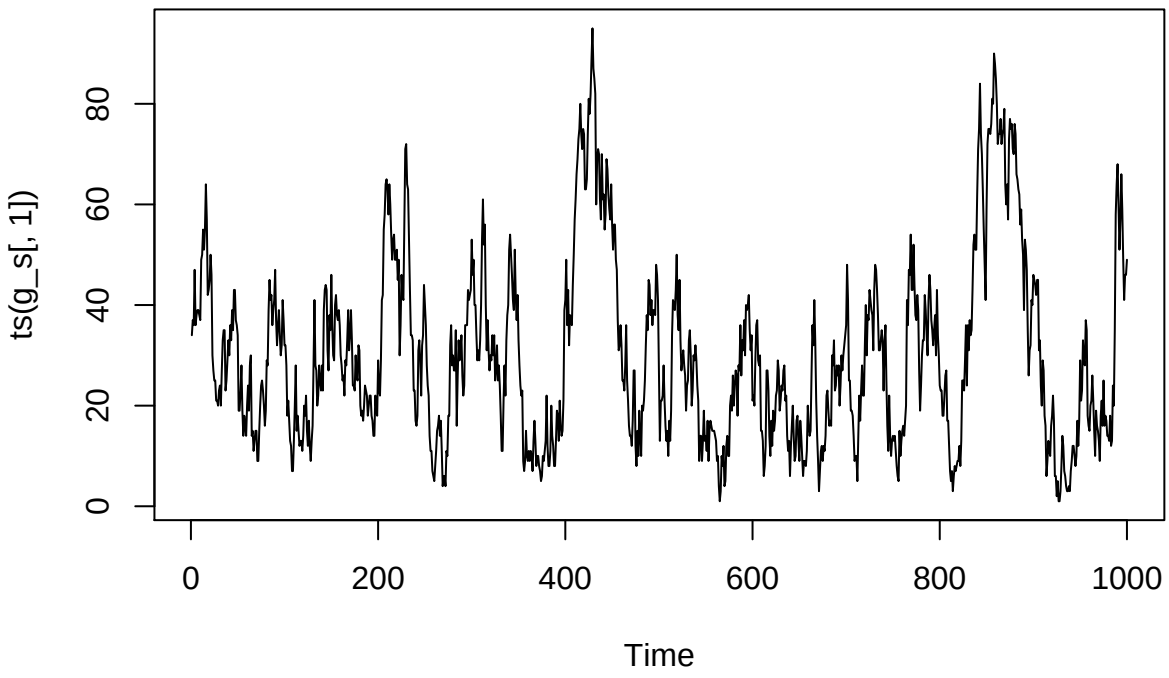
```
gib = function(a, b, n, N, x0, y0) {
## n:       , N:     , x0,y0:   , a,b,n:     .
gs <- matrix(ncol = 2, nrow = N) #N = length of chain
x <- x0 #initialize
y <- y0
gs[1, ] <- c(x, y)
for (i in 2:N) {
x <- rbinom(1, n, prob=y) # f(x|y)~bin(n,y)
y <- rbeta(1, x+a, n-x+b) # f(y|x)~beta(x+a, n-x+b)
gs[i, ] <- c(x, y)
}
gs

}
g_s<-gib(2, 4, 100, 101000, 2, 0.1)
g_s<-g_s[100001:101000,]
plot(g_s,col=1:1000)
```



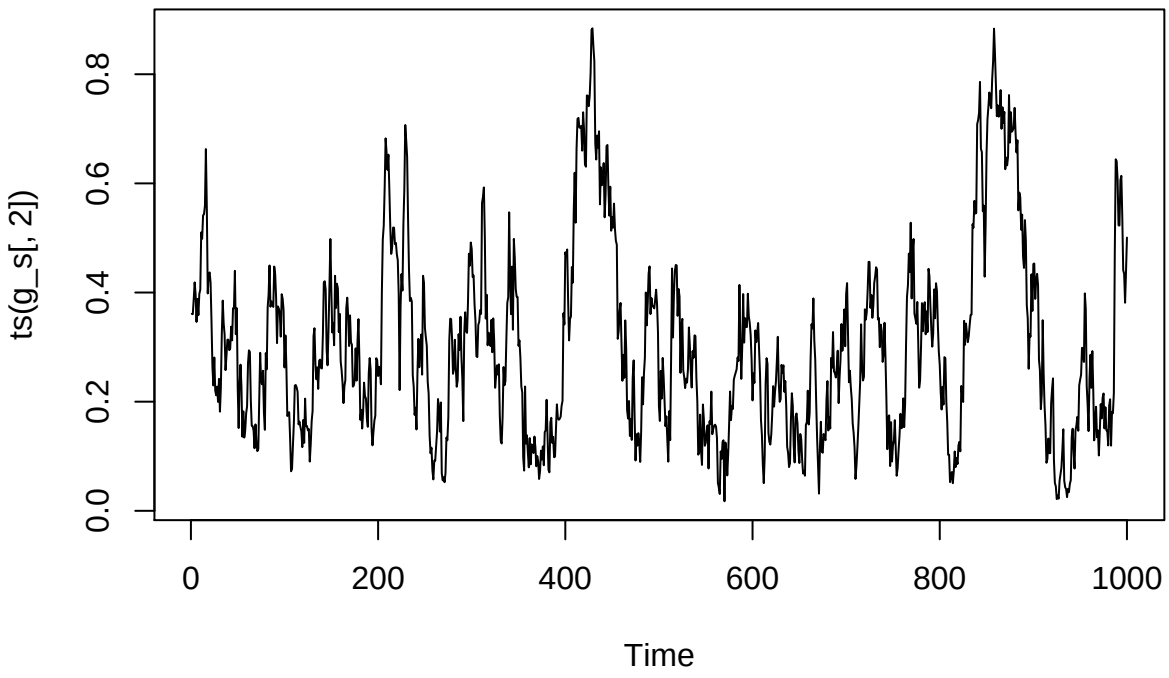

```
plot(g_s,type="l")
```



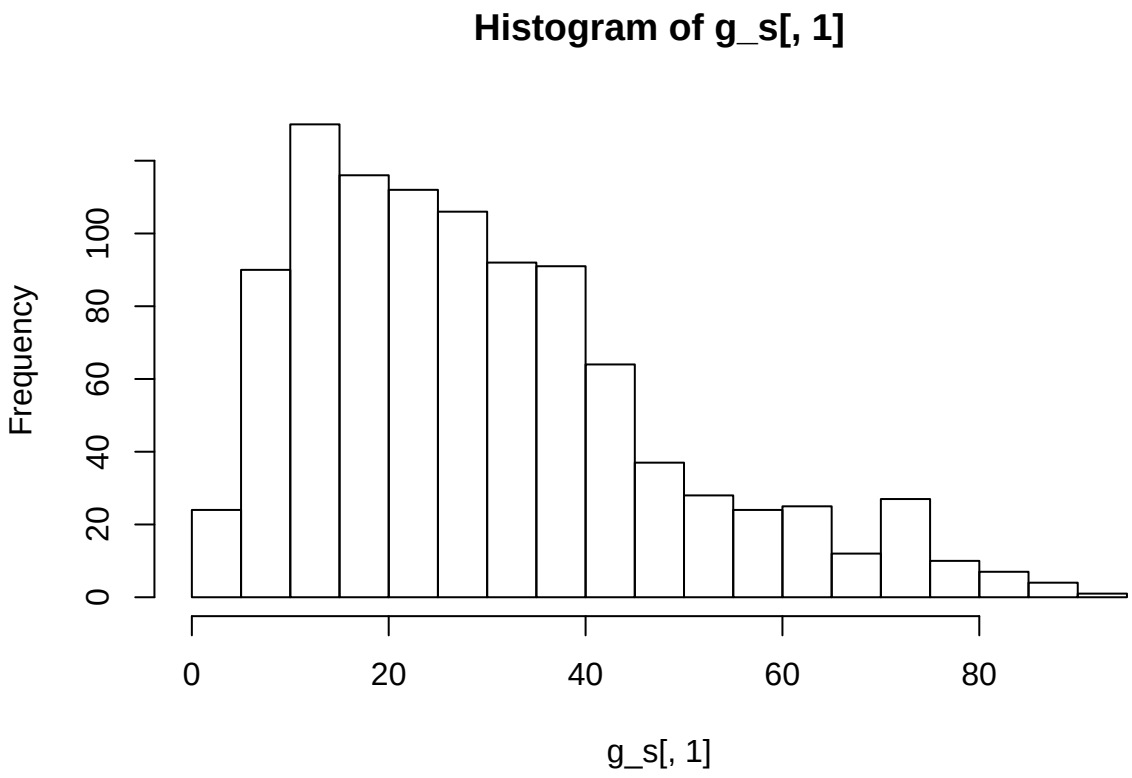

```
plot(ts(g_s[,1]))
```



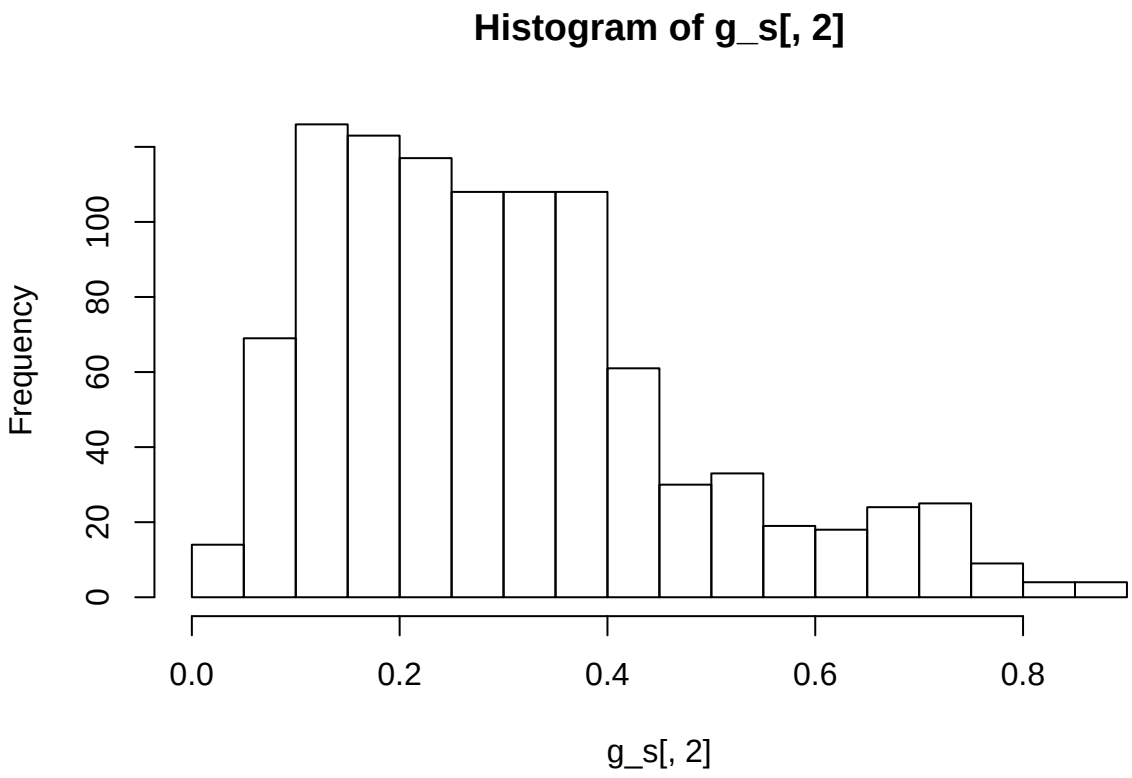

```
plot(ts(g_s[,2]))
```



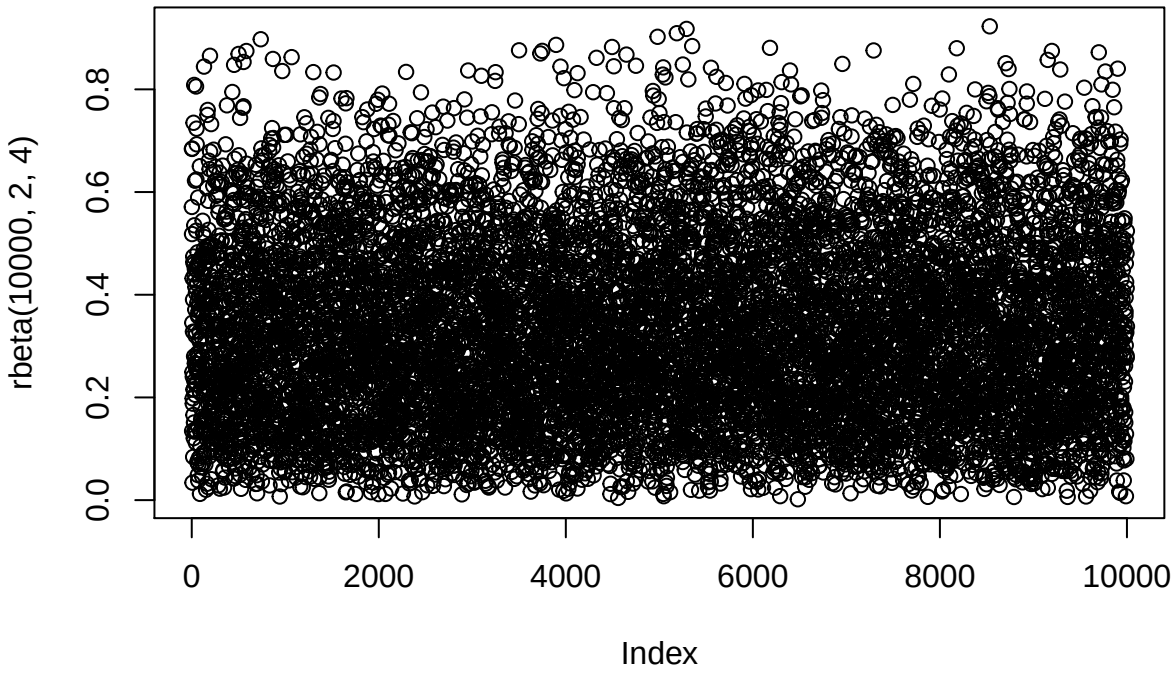

```
hist(g_s[,1],20)
```




```
hist(g_s[,2],20)
```




```
par(mfrow=c(1,1))
plot(rbeta(10000,2,4))
```



5.2 결론

주어진 함수를 따른다고 할 수 없다.

6 문제 7.A

Conduct a Monte Carlo study to estimate the coverage probabilities of the standard normal bootstrap confidence interval, the basic bootstrap confidence interval, and the percentile confidence interval. Sample from a normal population and check the empirical coverage rates for the sample mean. Find the proportion of times that the confidence intervals miss on the left, and the proportion of times that the confidence intervals miss on the right.

```
library(boot) #for boot and boot.ci
data<-rnorm(100)
theta.boot <- function(dat, i) {#function to compute the sample mean
y<-dat[i]
mean(y)
}
dat <- data
boot.obj <- boot(dat, statistic = theta.boot, R = 1000)# 1000      bootstrap
print(boot.obj)
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
```

```
## boot(data = dat, statistic = theta.boot, R = 1000)
##
##
## Bootstrap Statistics :
##      original        bias    std. error
## t1* 0.1930005 -0.0005091443   0.1052737
```

```
print(boot.ci(boot.obj, type = c("basic", "norm", "perc")))#calculations for bootstrap confidence inter
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 1000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = boot.obj, type = c("basic", "norm", "perc"))
##
## Intervals :
## Level      Normal              Basic              Percentile
## 95%   (-0.0128,  0.3998 )   ( 0.0009,  0.3898 )   (-0.0038,  0.3851 )
## Calculations and Intervals on Original Scale
```

```
alpha <- c(.025, .975) ## 95%
##                        .

#(1) normal
print(boot.obj$t0 + qnorm(alpha) * sd(boot.obj$t))
```

```
## [1] -0.01333205  0.39933309
```

```
#(2) basic
basicresult=2*boot.obj$t0 - quantile(boot.obj$t, rev(alpha), type=1)
print(basicresult) #rev
```

```
##       97.5%        2.5%
## 0.002294188 0.389816828
```

```
#(3) percentile
perresult=quantile(boot.obj$t, alpha, type=6)
print(perresult)
```

```
##         2.5%        97.5%
## -0.003792188  0.385097984
```

```
da<-boot.array(boot.obj,indices=T)
md<-numeric(100)
dda<-matrix(0,100,1000)
for(i in 1:1000){
dda[,i]<-da[i,]
}
for(i in 1:100){
for(j in 1:1000)
dda[i,j]<-dat[dda[i,j]]
}
md<-colMeans(dda)

#
(sum(md<(-0.1579)))/1000; (sum(md>0.2405))/1000
```

```
## [1] 0
```

```
## [1] 0.333
```

```
(sum(md<(-0.1621)))/1000; (sum(md>0.2436))/1000
```

```
## [1] 0
```

```
## [1] 0.318
```

```
(sum(md<(-0.1466)))/1000; (sum(md>0.2610))/1000
```

```
## [1] 0.001
```

```
## [1] 0.269
```

6.1 결과

proportion of times that the confidence intervals miss를 구하기 위해 boot함수에서 제공하는 boot.array를 통해 bootstrap이 일어난 변수들을 구해서 알아본 결과이다. 수행결과를 보면 오른쪽으로 꼬리가 긴 카이제곱형태를 따르는 것 처럼 보인다.