407 Assignment 2

Haohan Yao

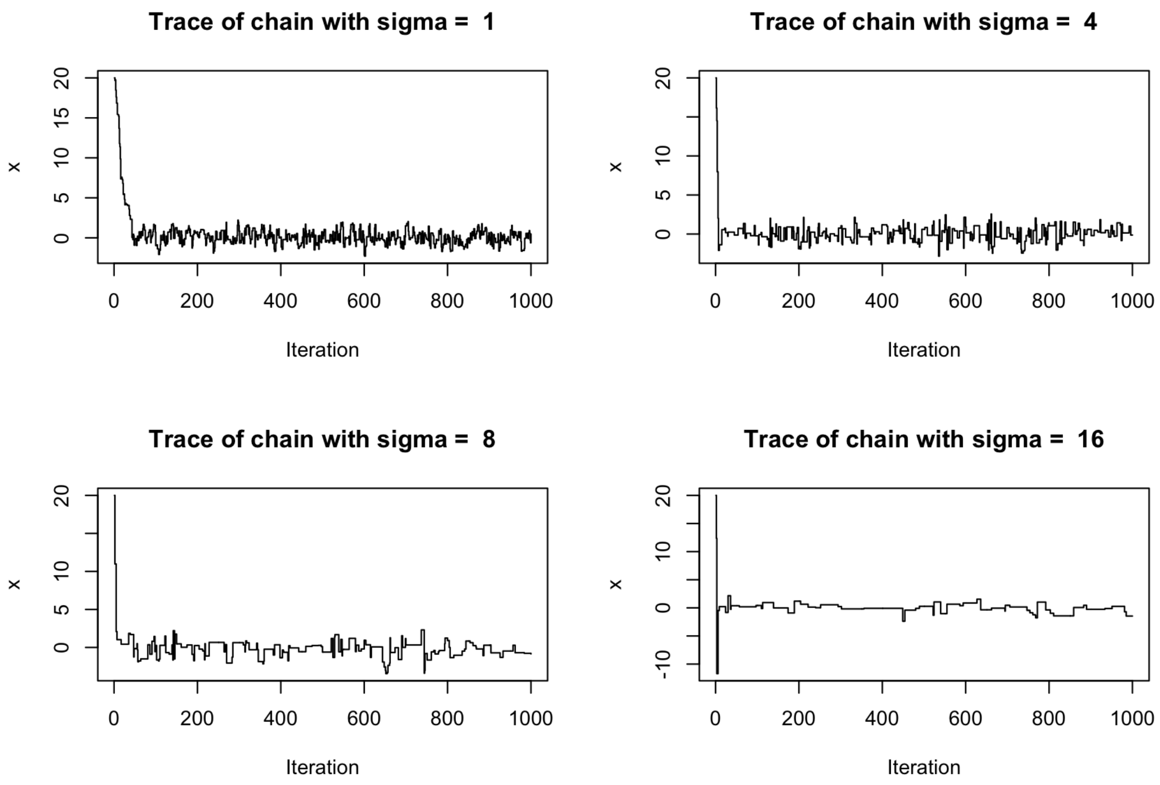
u2119251

question 1

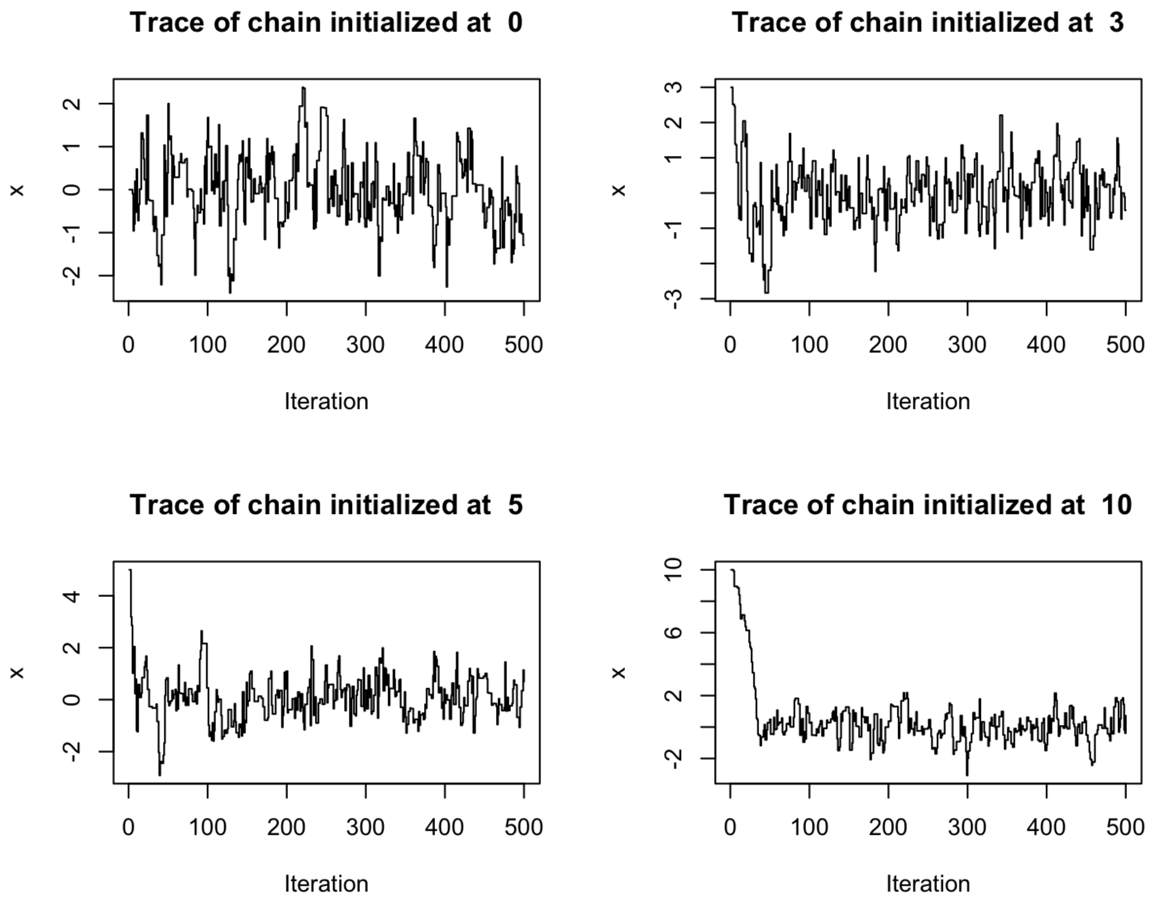
```
metrp = function(n=1000,Sigma = 1,miu = 0,x0=0,Beta = 1.5){
  X = matrix(0,nrow = 1,ncol = n)
  X[1] = x0
  U = runif(n)
  for (i in 2:n) {
    Y = X[i-1]+rnorm(1,mean = miu,sd = Sigma)
#    alpha = exp((abs((X[i-1]-miu)/Sigma))^Beta-(abs((Y-miu)/Sigma))^Beta)
    alpha = exp(abs(X[i-1])^Beta-abs(Y)^Beta)
    if(U[i]<alpha){
      X[i] = Y
    }
    else{
      X[i] = X[i-1]
    }
  }
  X
}
```

(a)

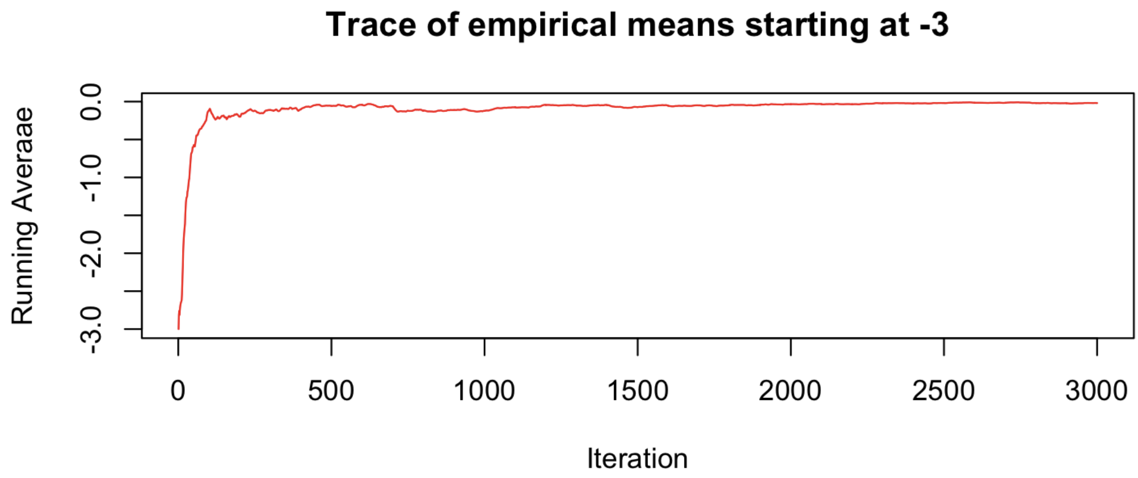
i. the choice of Sigma have influence on the acceptance rate, I would like to choose Sigma=1

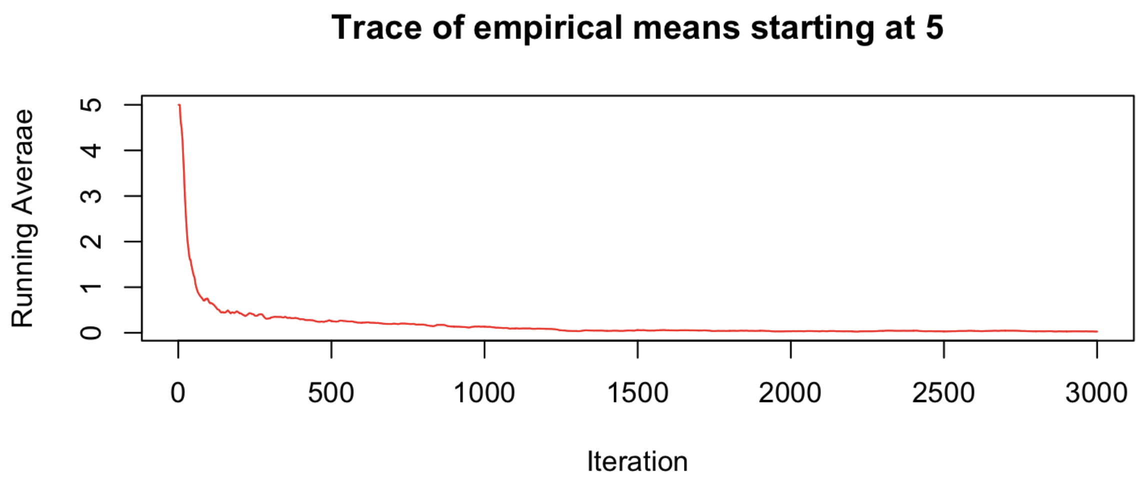


ii. the initial X_0 would have influence on the $Burn-in$ process, through these plots we can see that the sequence will finally fluctuate around 0, so we can decide by the stablization of the chain



iii. I plot the empirical means of a chain through iteration, it become so steady around 0.





(b)

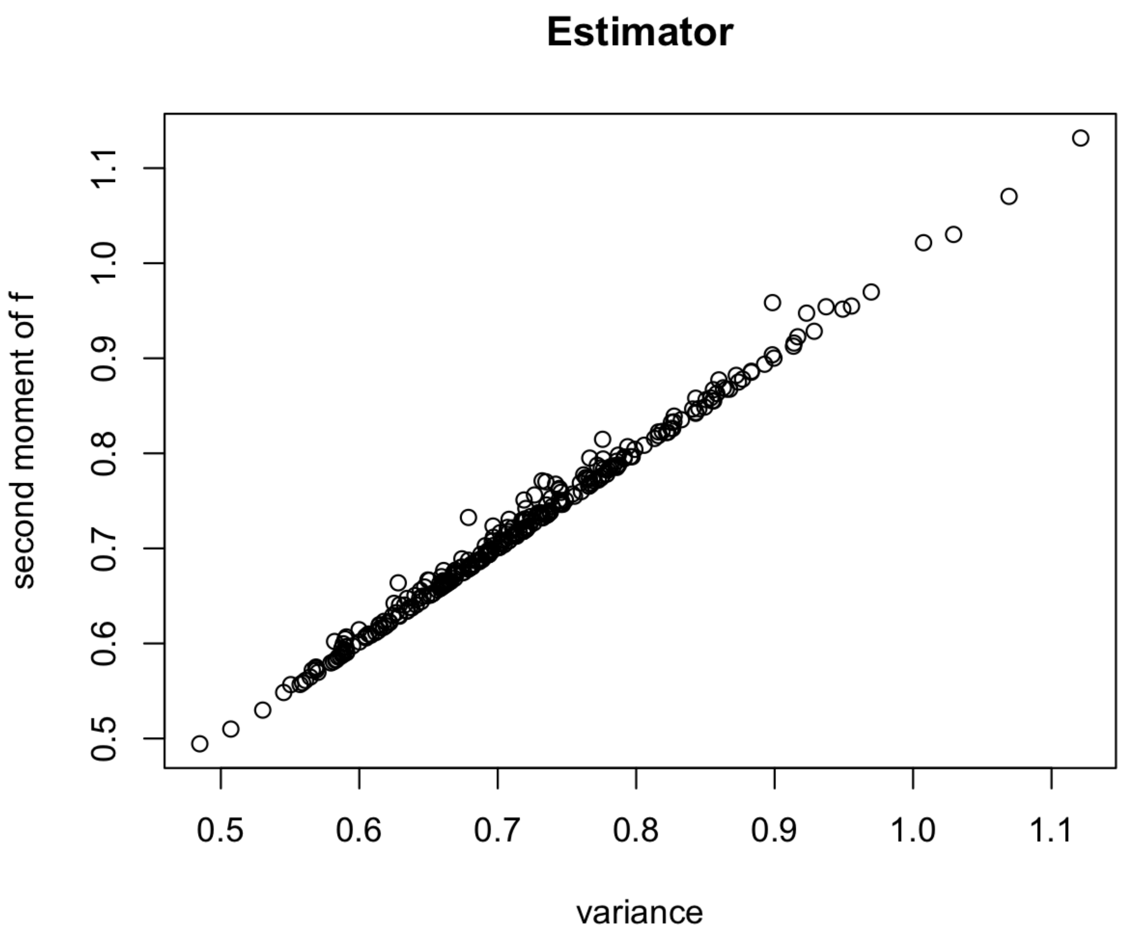
```
est = c()
for (i in 1:300) {
  xb = metrp(x0=0,Sigma=1 ,n = 1000)
  est[i] = mean(xb^2)
}
var(est)
```

$$var(Estimator) = 0.0114115$$

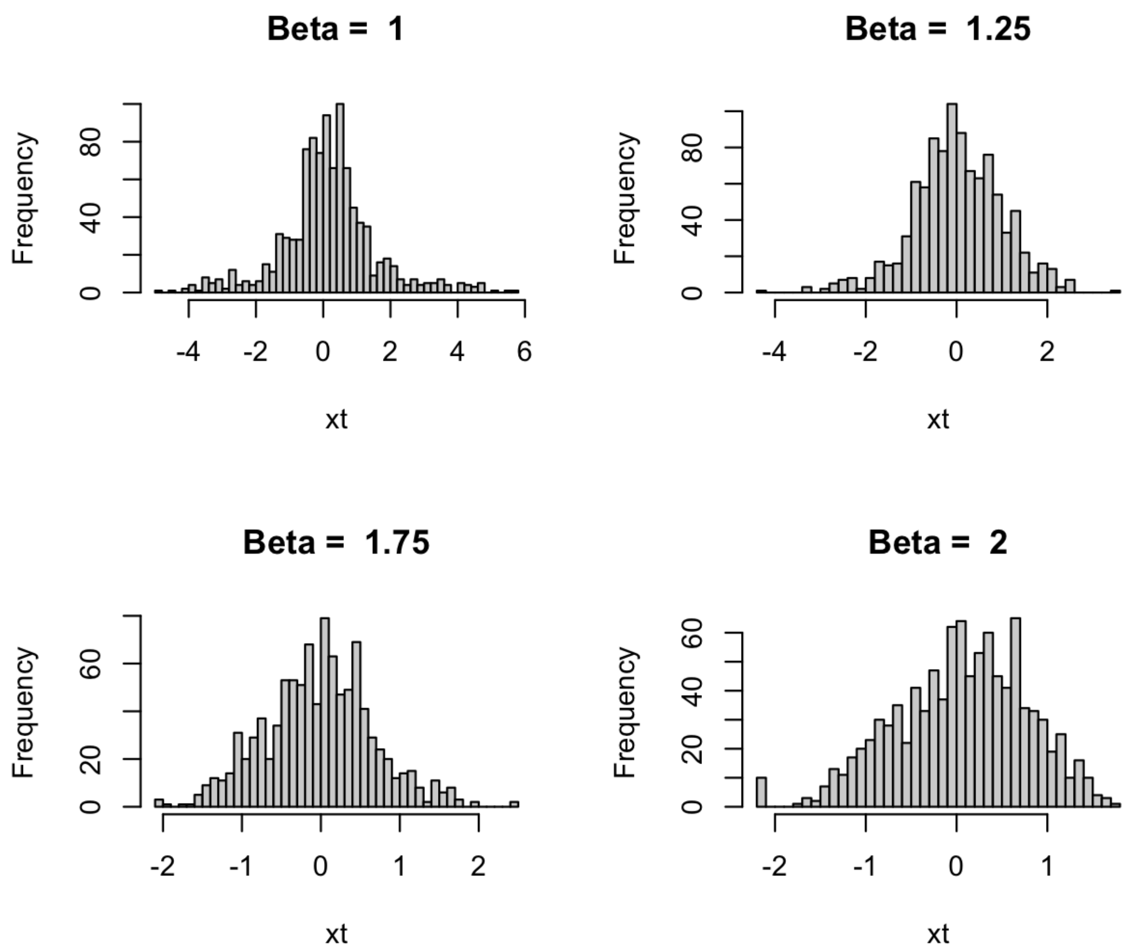
(c)

as the mean of distribution is 0, so the variance of f is equal to $second\ moment\ of\ f$

so we can estimate like part b. to calculate the variance.



invariant distribution of Markov chain for different β



Possibly, as we have known the combination distribution of $X \ and \ \beta$, we could calculate the conditional distribution and then apply $Gibbs \ sampling$

question 2

```
pdf = function(x){
  if(x>-5 && x<(-3)){
    return(3/8*(1-(x+4)^2))
  }
  else if(x>3 && x<5){
    return(3/8*(1-(x-4)^2))
  }
  else return(0)
}
```

(a)

(a) through the sample steps, it's obvious that

① $p(u_t \mid x_{t-1}) = U(0, f(x_{t-1}))$

② $p(x_t \mid u_t) = U(x : f(x) > u_t)$

$p(x_t, u_t) = P(x_{t-1}, u_{t-1}) \cdot p(u_t \mid x_{t-1}) \cdot p(x_t \mid u_t)$

it corresponds to Gibbs Sampling Steps

Starting at $x^{(0)}$ and through the 2 steps, the sample area is limited into $\{(x, u) : u < f(x)\}$,

(b)

(b). $p(x) = \int_u p(x, u)\, du$

$= \int_0^{f(x)} \frac{1}{Z}\, du$

$= \frac{f(x)}{Z}$, $Z = \int f(x)\, dx = 1$

$\therefore$ the invariant distribution is $f(x)$

(c)

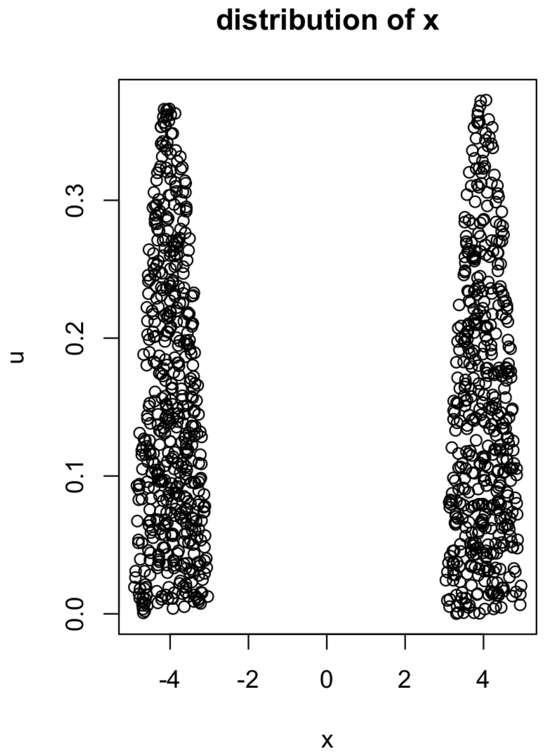
1. Draw $U^{(t)} \sim U(0, f(x_{(t-1)}))$

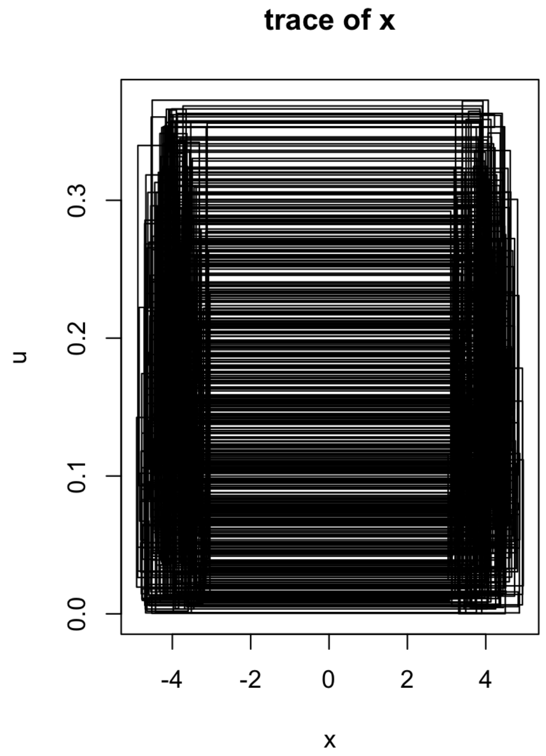
2. Draw $X^{(t)} \sim$

$$U\left\{x : x \in \left[-4-\sqrt{1-\frac{8}{3}u_t},\ -4+\sqrt{1-\frac{8}{3}u_t}\right] \cup \left[4-\sqrt{1-\frac{8}{3}u_t},\ 4+\sqrt{1-\frac{8}{3}u_t}\right]\right\}$$

slice sampler:

```
slice = function(n=1000,x0=as.vector(c(0,0))){
  X = matrix(0,nrow = 2,ncol = n)
  X[,1] = x0
  for (i in 2:n) {
   ###U(t)
    X[1,i] = runif(1,0,pdf(X[2,i-1]))
   ###X(t)
    X[2,i] = runif(1,-sqrt(1-X[1,i]*8/3),sqrt(1-X[1,i]*8/3))+(runif(1,-1,1)>0)*8-4
  }
  X
}
par(mfrow = c(1,2))
xss = slice(x0 = as.vector(c(0.1,4)))
plot(xss[2,],xss[1,],xlab = 'x',ylab = 'u',main ='distribution of x')
plot(xss[2,],xss[1,],type = 's',xlab = 'x',ylab = 'u',main='trace of x')
```





Metropolis algorithm

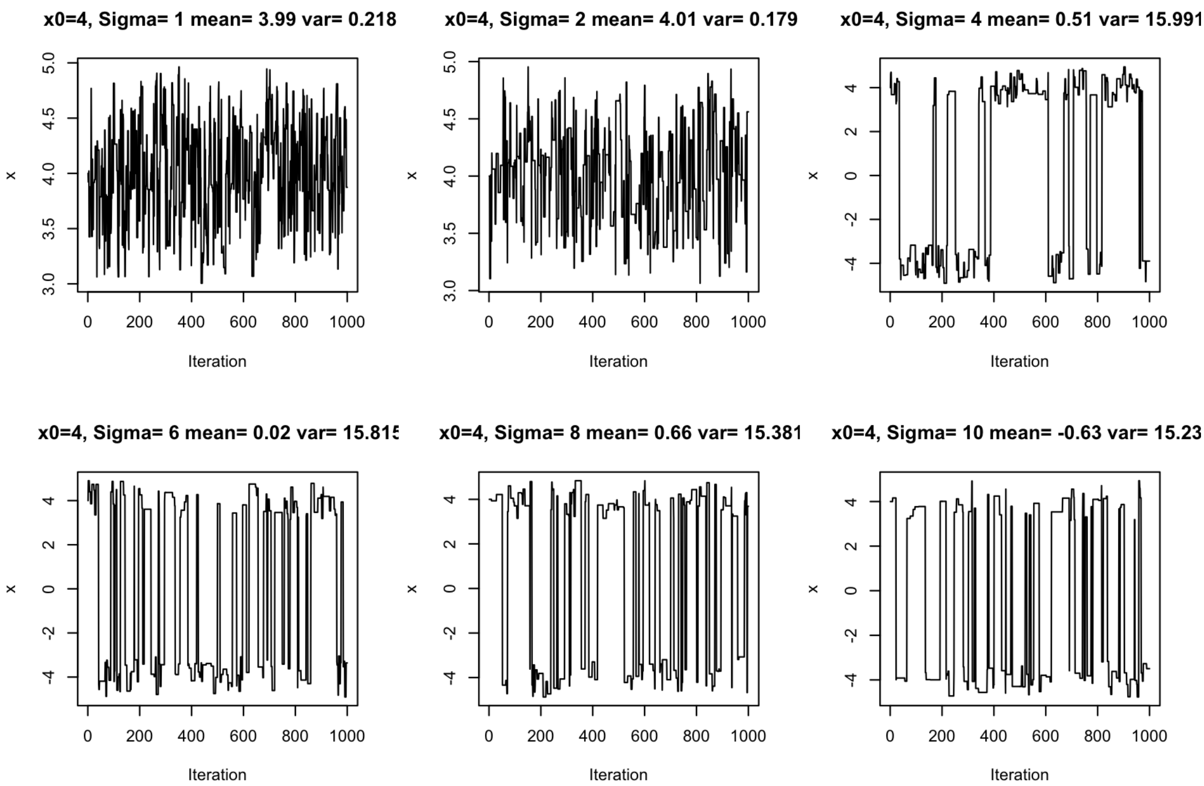
```
#####metropolis
mh = function(n=1000,Sigma = 1,miu = 0,x0=0,Beta = 1.5){
  X = matrix(0,nrow = 1,ncol = n)
  X[1] = x0
  U = runif(n)
  for (i in 2:n) {
    Y = X[i-1]+rnorm(1,mean = miu,sd = Sigma)
    alpha = pdf(Y)/pdf(X[i-1])
    if(U[i]<alpha){
      X[i] = Y
    }
    else{
      X[i] = X[i-1]
    }
  }
  X
}

par(mfrow = c(2,3))
xm = matrix(0,nrow = 8,ncol = 1000)
count = 1
for(i in c(1,2,4,6,8,10)){
  xm[count,] = mh(Sigma = i,x0 = 4)
  m = round(mean(xm[count,]),2)
  v = round(var(xm[count,]),3)
  name = paste(' x0=4, Sigma=',i,'mean=',m,'var=',v)
  plot(seq(1,1000,1),xm[count,],type = 'l',xlab = 'Iteration',ylab ='x',main = name )
```

```
  count = count + 1
}
```

we choose different Sigma and calculate the mean and var for every Sigma

plot the trace as following :



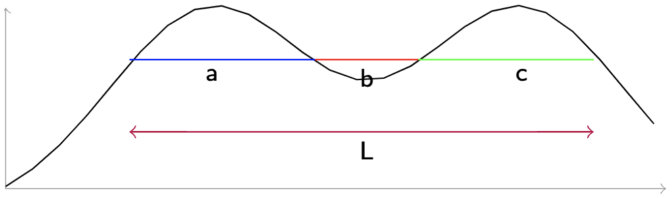
obviously there is an huge change of Mean and Var through the increase of Sigma, especially from Sigma = 2 to 4

Through the plots of two algorithms I think the quality of slice sampling is better as Metropolis algorithm could be influence by choice of Sigma, however using Slice algorithm request to choose a proper initial X_0 that $f(X_0) > 0$

(d)

i) Slice sampling is hard to implement for high dimensions because we when we choose X_t , we need to make sure that X_t~$U\{x_t : f(x_t) >= U_t\}$, but in real problems the area of this is often divided into many different parts depending on U_t ,and when it comes to high dimension it tends to be multi_dimension plates, which makes it harder to draw X_t uniformly.

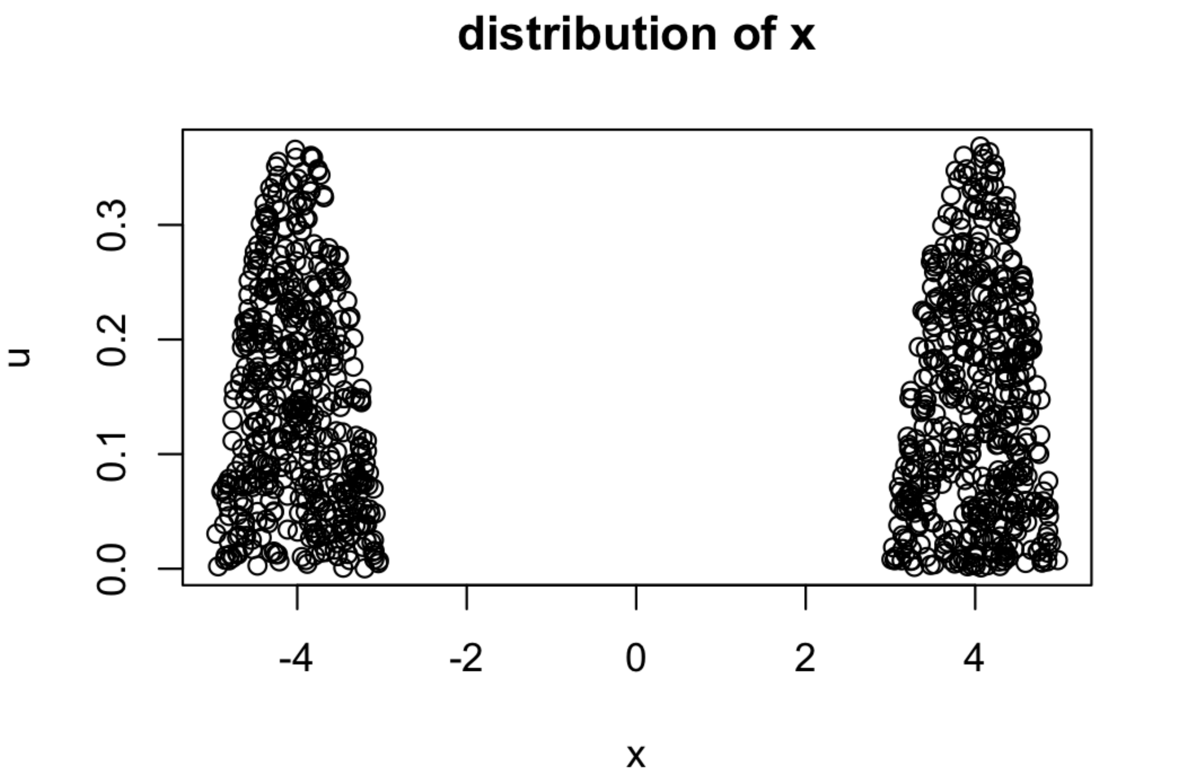
ii) we can use Shrinkage Algorithm in order to deal with the gaps between the area of X_t ,



$$\Pr(X^{(t)} \in a \to X^{(t+1)} \in a) = \frac{a+b}{L}$$
$$\Pr(X^{(t)} \in a \to X^{(t+1)} \in c) = \frac{c}{L}$$
$$\Pr(X^{(t)} \in c \to X^{(t+1)} \in a) = \frac{a}{L}$$
$$\Pr(X^{(t)} \in c \to X^{(t+1)} \in c) = \frac{b+c}{L}$$

Therefore,

$$\begin{aligned}\Pr(X^{(t+1)} \in a) &= \Pr(X^{(t)} \in a \to X^{(t+1)} \in a)\Pr(X^{(t)} \in a) + \Pr(X^{(t)} \in c \to X^{(t+1)} \in a)\Pr(X^{(t)} \in c) \\ &= \frac{a+b}{L}\frac{a}{L} + \frac{a}{L}\frac{c}{L} = \frac{a^2+ab+ac}{L^2} = \frac{a}{L} = \Pr(X^{(t)} \in a)\end{aligned}$$



iii)

I think the Shrinkage algorithm add a estimate to the Slice sampler which is like the way that Metropolis Hasting use: to change the probability of transforming depending on the different area of X_t in order to make sure the

$Detailed\ Balance\ Function$