Examples of Samplers

Shunri Zheng

2023-04-06

```
library(ggplot2)
library(HDInterval)
library(invgamma)
library(combinat)
```

```
##
## Attaching package: 'combinat'

## The following object is masked from 'package:utils':
##
##     combn
```

```
library(MASS)
library(MCMCpack)
```

```
## Loading required package: coda

## ##
## ## Markov Chain Monte Carlo Package (MCMCpack)

## ## Copyright (C) 2003-2023 Andrew D. Martin, Kevin M. Quinn, and Jong Hee Park

## ##
## ## Support provided by the U.S. National Science Foundation

## ## (Grants SES-0350646 and SES-0350613)
## ##

##
## Attaching package: 'MCMCpack'

## The following objects are masked from 'package:invgamma':
##
##     dinvgamma, rinvgamma
```

```
library(emdbook)
library(mvtnorm)
```

```
##
## Attaching package: 'mvtnorm'

## The following object is masked from 'package:emdbook':
##
##     dmvnorm
```

κ

Question 4

(a)

Since $0 \le |x-y| < c$, WLOG, we only need to discuss the three cases:

1. $0 < c \le 0.5$
2. $0.5 < c < 1$
3. $c \ge 1$

Case 1: $0 < c \le 0.5$

The conditional pdf $f(y|x)$ is:

$$f(y|x) = \frac{f(x,y)}{f(x)} = \begin{cases} \dfrac{1}{\int_0^{x+c} dy} = 1/(x+c)\mathbb{I}_{0 \le y \le x+c}, & \text{if } 0 \le x \le c \\ \dfrac{1}{\int_{x-c}^{x+c} dy} = 1/(2c)\mathbb{I}_{x-c \le y \le x+c}, & \text{if } c \le x \le 1-c \\ \dfrac{1}{\int_{x-c}^{1} dy} = 1/(1-x+c)\mathbb{I}_{x-c \le y \le 1}, & \text{if } 1-c \le x \le 1 \end{cases} \tag{1}$$

Similarly, the conditional pdf $f(x|y)$ is:

$$f(x|y) = \frac{f(x,y)}{f(y)} = \begin{cases} \dfrac{1}{\int_0^{y+c} dx} = 1/(y+c)\mathbb{I}_{0 \le x \le y+c}, & \text{if } 0 \le y \le c \\ \dfrac{1}{\int_{y-c}^{y+c} dx} = 1/(2c)\mathbb{I}_{y-c \le x \le y+c}, & \text{if } c \le y \le 1-c \\ \dfrac{1}{\int_{y-c}^{1} dx} = 1/(1-y+c)\mathbb{I}_{y-c \le x \le 1}, & \text{if } 1-c \le y \le 1 \end{cases} \tag{2}$$

Case 2: $0.5 < c < 1$

The conditional pdf $f(y|x)$ is:

$$f(y|x) = \frac{f(x,y)}{f(x)} = \begin{cases} \dfrac{1}{\int_0^{x+c} dy} = 1/(x+c)\mathbb{I}_{0 \le y \le x+c}, & \text{if } 0 \le x \le 1-c \\ \dfrac{1}{\int_0^{1} dy} = \mathbb{I}_{0 \le y \le 1}, & \text{if } 1-c \le x \le c \\ \dfrac{1}{\int_{x-c}^{1} dy} = 1/(1-x+c)\mathbb{I}_{x-c \le y \le 1}, & \text{if } c \le x \le 1 \end{cases} \tag{3}$$

Similarly, the conditional pdf $f(x|y)$ is:

$$f(x|y) = \frac{f(x,y)}{f(y)} = \begin{cases} \dfrac{1}{\int_0^{y+c} dx} = 1/(y+c)\mathbb{I}_{0 \le x \le y+c}, & \text{if } 0 \le y \le 1-c \\ \dfrac{1}{\int_0^{1} dx} = \mathbb{I}_{0 \le x \le 1}, & \text{if } 1-c \le y \le c \\ \dfrac{1}{\int_{y-c}^{1} dx} = 1/(1-y+c)\mathbb{I}_{y-c \le x \le 1}, & \text{if } c \le y \le 1 \end{cases} \tag{4}$$

Case 3: $c \ge 1$

The conditional pdf $f(y|x)$ is:

$$f(y|x) = \frac{f(x,y)}{f(x)} = \mathbb{I}_{0 \le y \le 1}, \text{if } 0 \le x \le 1 \tag{5}$$

Similarly, the conditional pdf $f(x|y)$ is:

$$f(x|y) = \frac{f(x,y)}{f(y)} = \mathbb{I}_{0 \le x \le 1}, \text{if } 0 \le y \le 1 \tag{6}$$

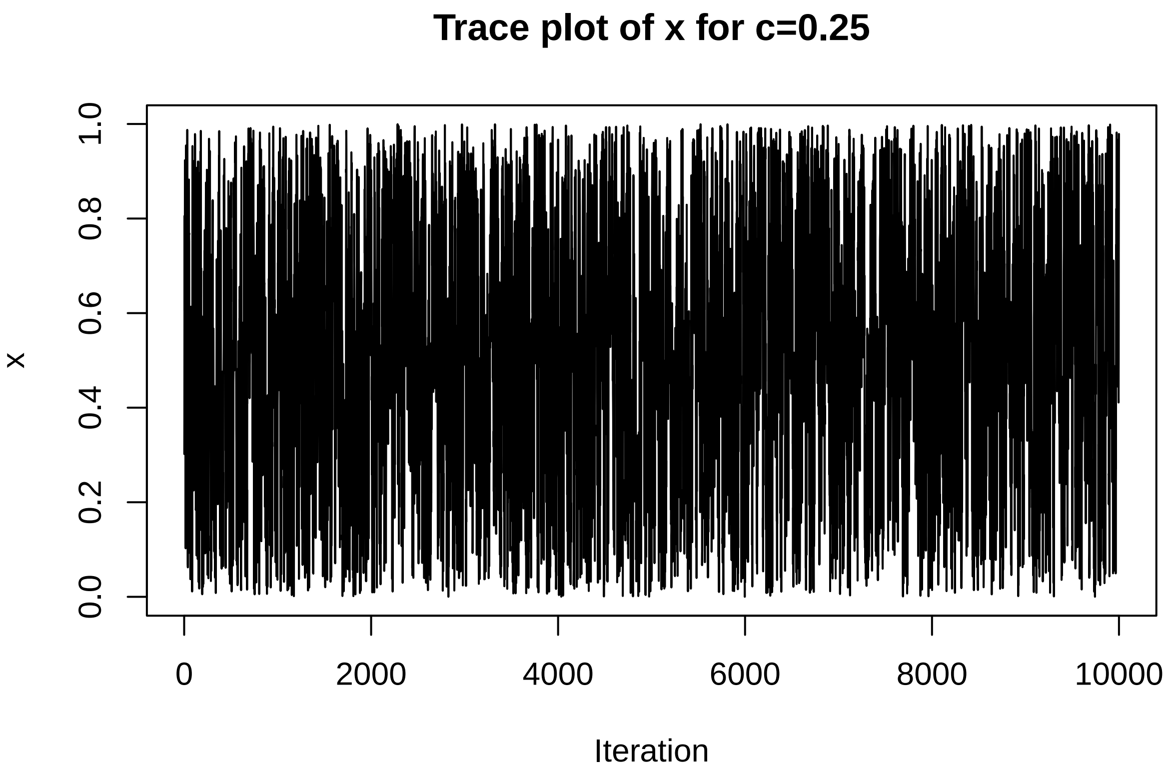
(b)

Gibbs sampler for f(x,y) is:

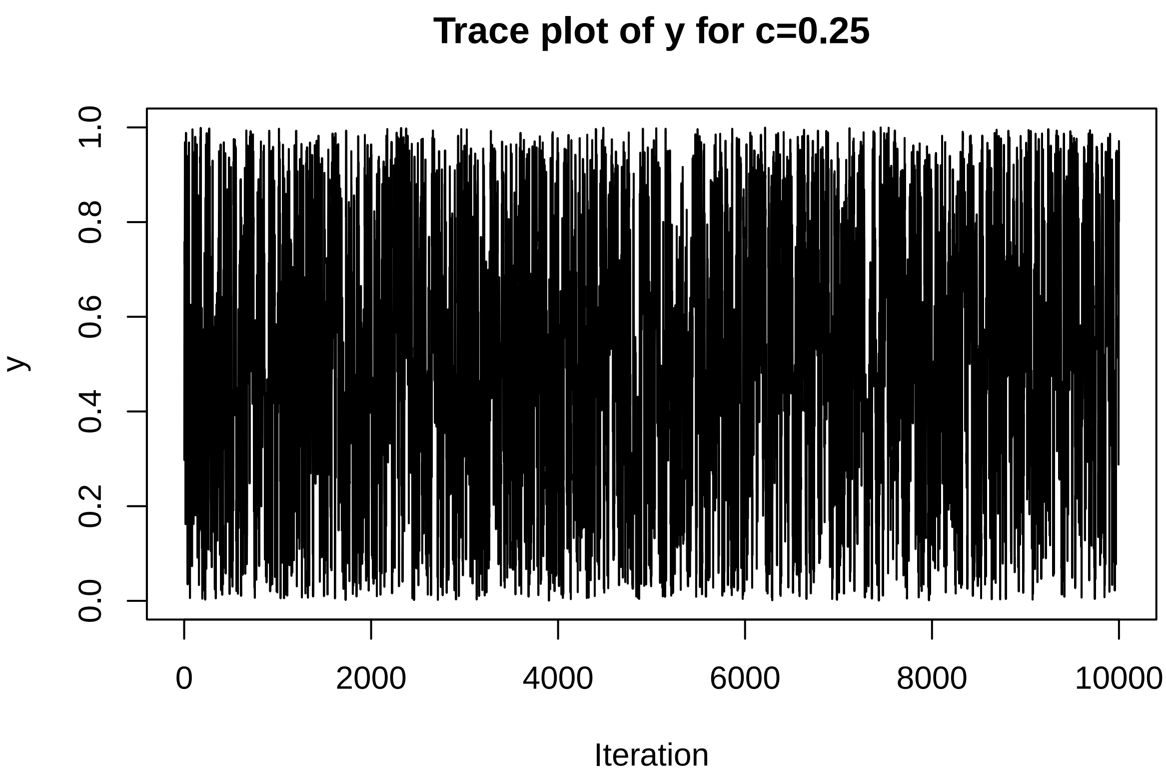
```
set.seed(20230225)
x_given_y <- function(y, c){
    if(y <= c){
        return(runif(1, 0, y+c))
    }else if(y <= 1-c){
        return(runif(1, y-c, y+c))
    }else{
        return(runif(1, y-c, 1))
    }
}

y_given_x <- function(x, c){
    if(x <= c){
        return(runif(1, 0, x+c))
    }else if(x <= 1-c){
        return(runif(1, x-c, x+c))
    }else{
        return(runif(1, x-c, 1))
    }
}


sampleGibbs <- function(start.a, start.b,c, n.sims,burnin=0){

# initialize the chain
chain <- matrix(NA, nrow=n.sims, ncol=2)
chain[1,] <- c(start.a, start.b)

# loop through the chain
for(i in 2:n.sims){
    chain[i,1] <- x_given_y(chain[i-1,2], c)
    chain[i,2] <- y_given_x(chain[i,1], c)
}

# return the chain
return(chain[(burnin+1):n.sims,])
}



Samples_From_Gibbs1 = sampleGibbs(0.5, 0.5,0.25, 11000, 1000)
Samples_From_Gibbs2 = sampleGibbs(0.5, 0.5,0.05, 11000, 1000)
Samples_From_Gibbs3 = sampleGibbs(0.5, 0.5,0.02, 11000, 1000)
```

(c)

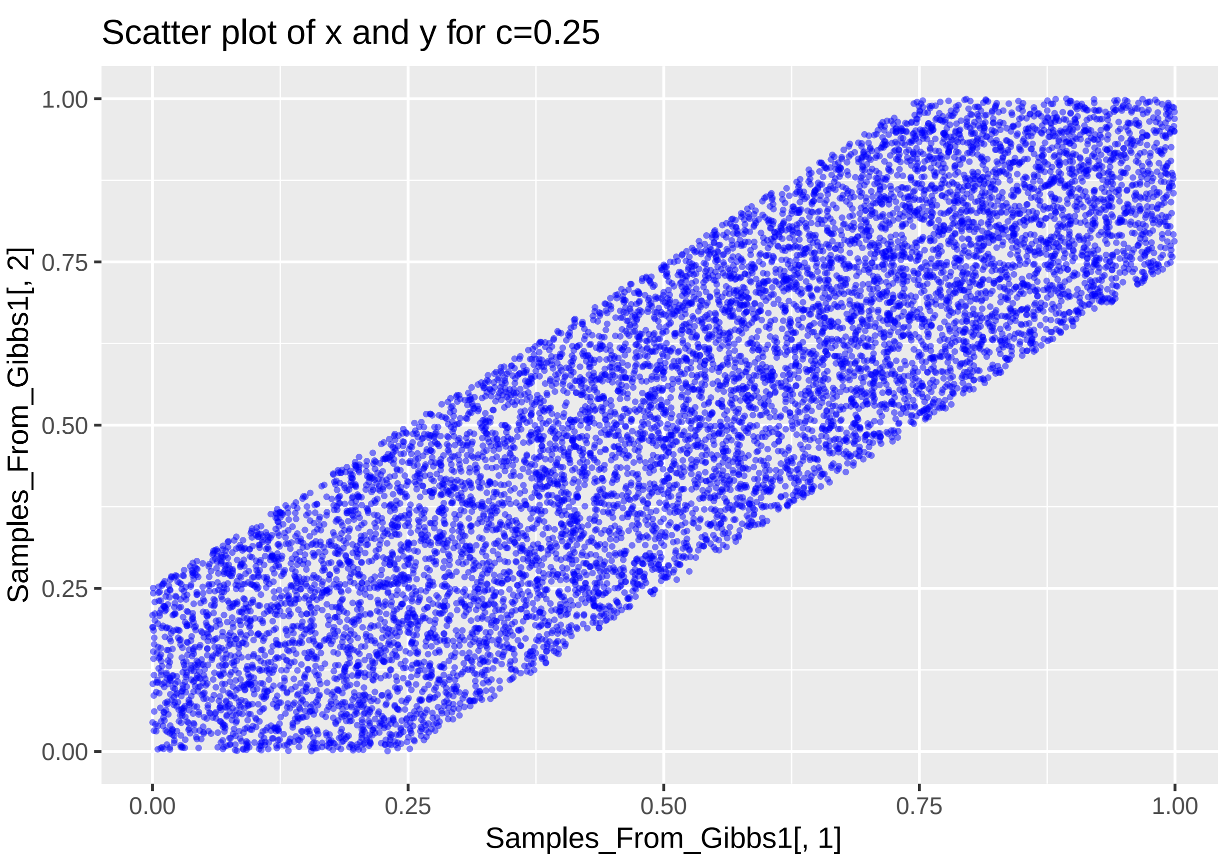
```
ts.plot(Samples_From_Gibbs1[,1], type="l", col="black", lwd=1, xlab="Iteration",
 ylab="x", main="Trace plot of x for c=0.25")
```




```
ts.plot(Samples_From_Gibbs1[,2], type="l", col="black", lwd=1, xlab="Iteration",
 ylab="y", main="Trace plot of y for c=0.25")
```

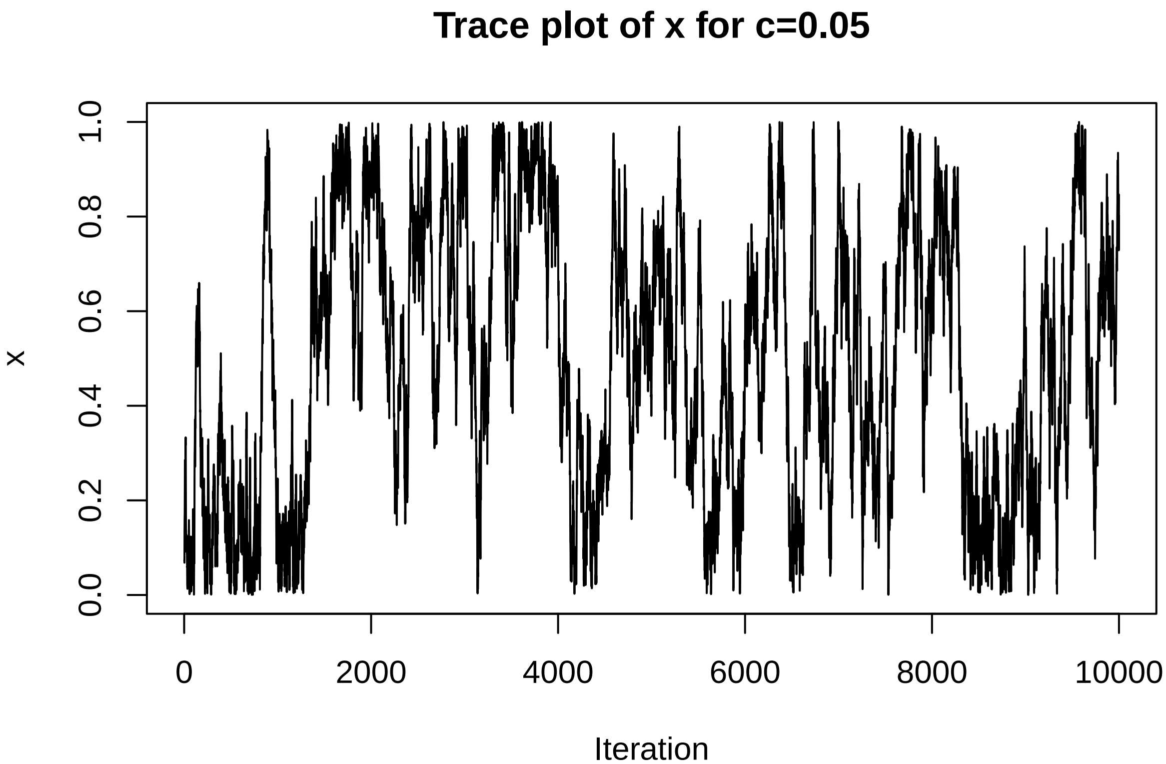



```
ggplot(data.frame(Samples_From_Gibbs1), aes(x = Samples_From_Gibbs1[,1],
 y = Samples_From_Gibbs1[,2])) + geom_point(alpha = 0.5,size = 0.5,color = "blue") + ggtitle("Scatter pl
```

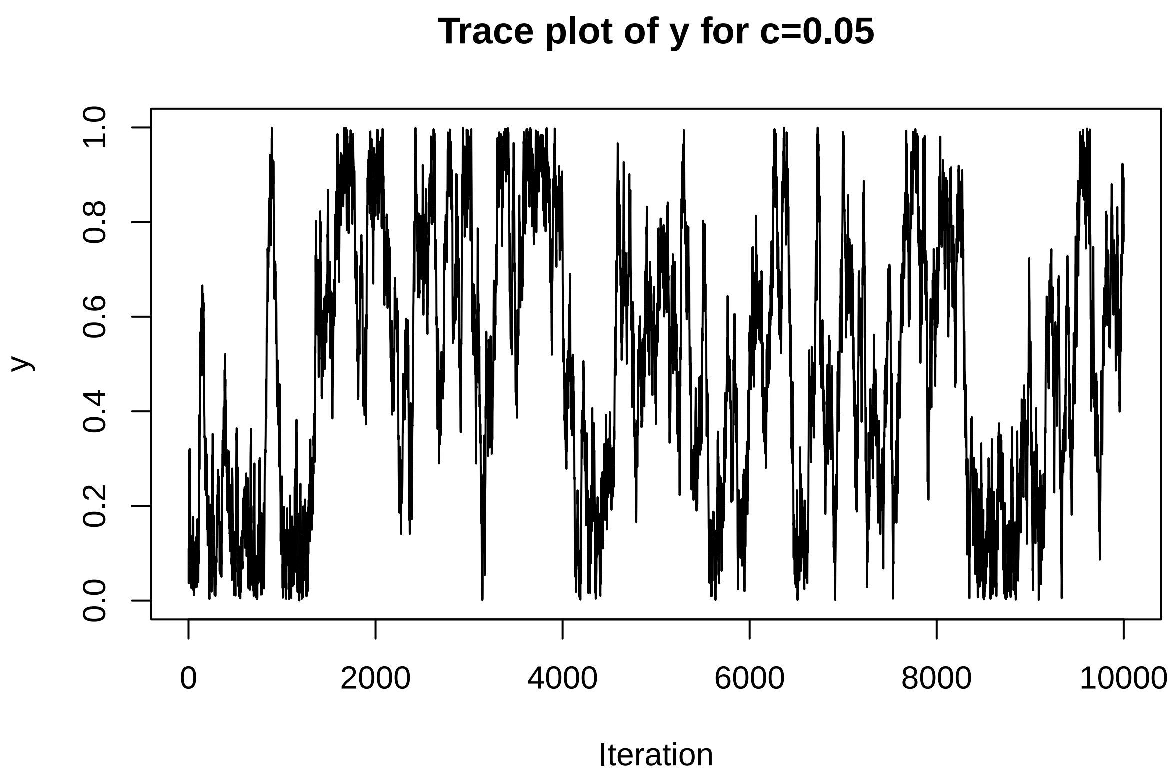


```r
par(mfrow=c(3,1))
```

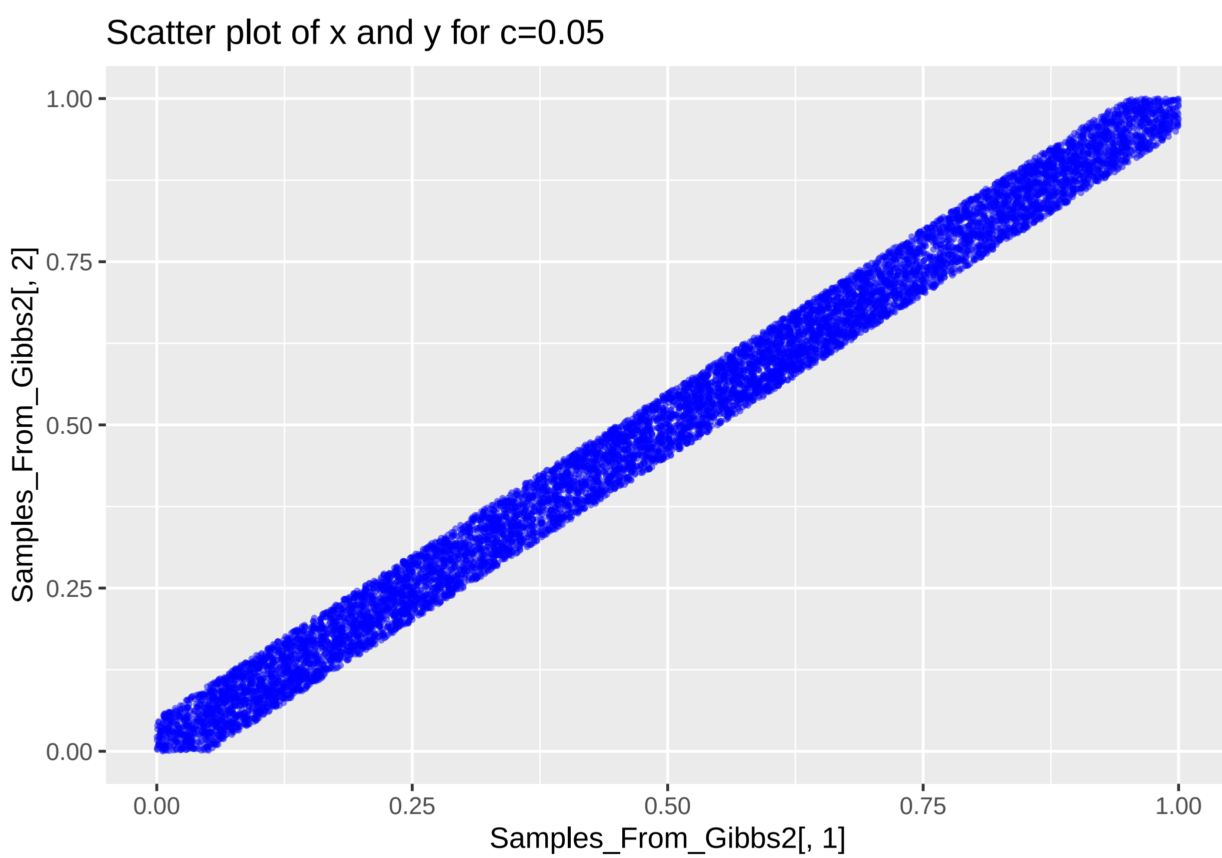
```r
ts.plot(Samples_From_Gibbs2[,1], type="l", col="black", lwd=1, xlab="Iteration",
 ylab="x", main="Trace plot of x for c=0.05")
```




```r
ts.plot(Samples_From_Gibbs2[,2], type="l", col="black", lwd=1, xlab="Iteration",
    ylab="y", main="Trace plot of y for c=0.05")
```

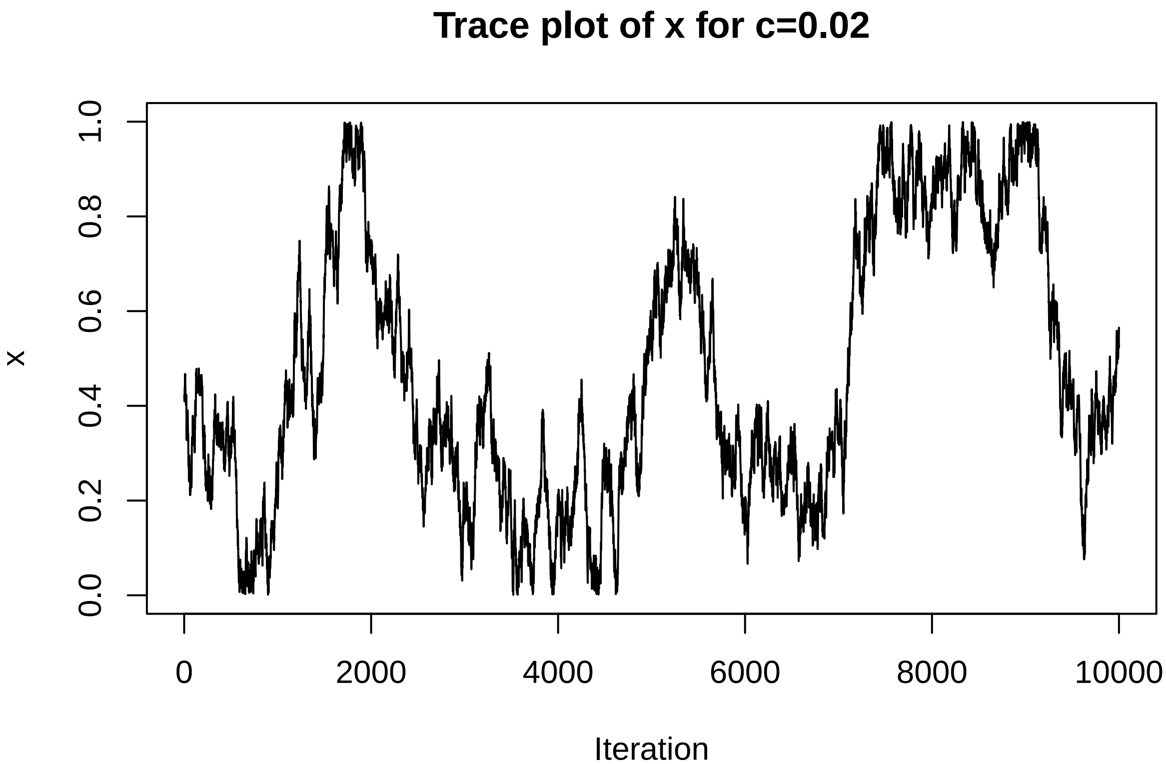



```
ggplot(data.frame(Samples_From_Gibbs2), aes(x = Samples_From_Gibbs2[,1],
    y = Samples_From_Gibbs2[,2])) + geom_point(alpha = 0.5,size = 0.5,color = "blue") + ggtitle("Scatte
```

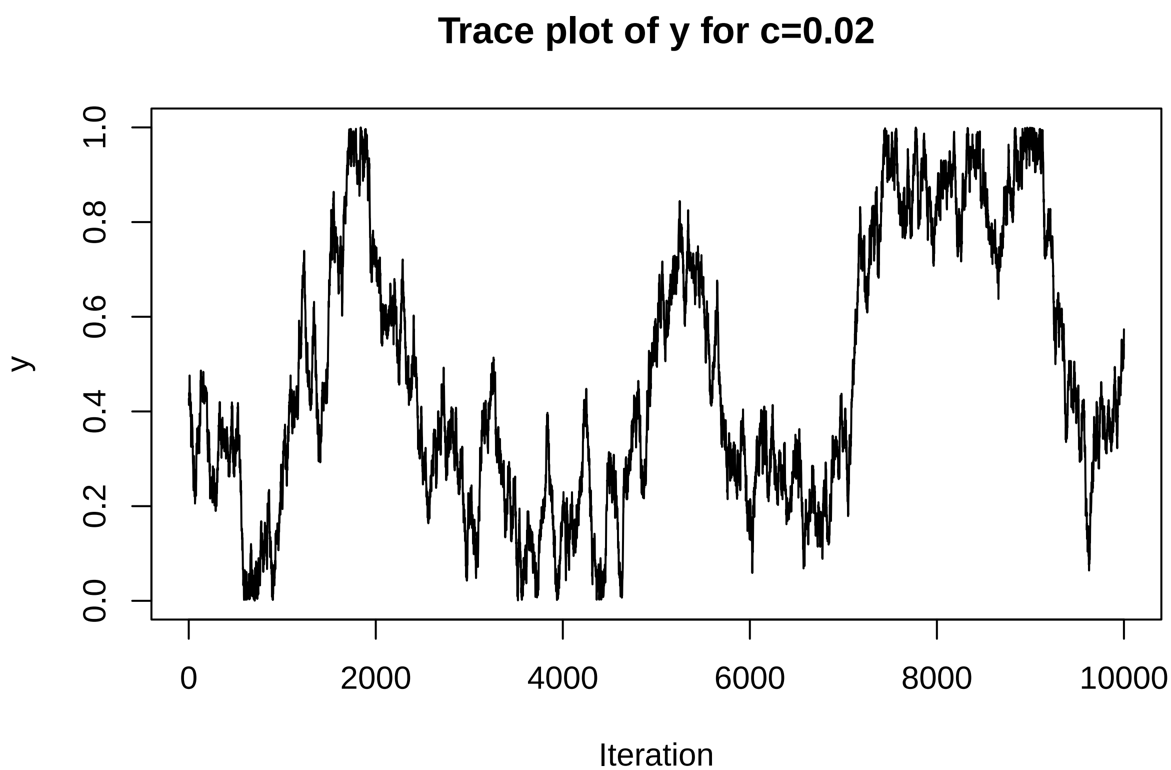


```
par(mfrow=c(3,1))
```

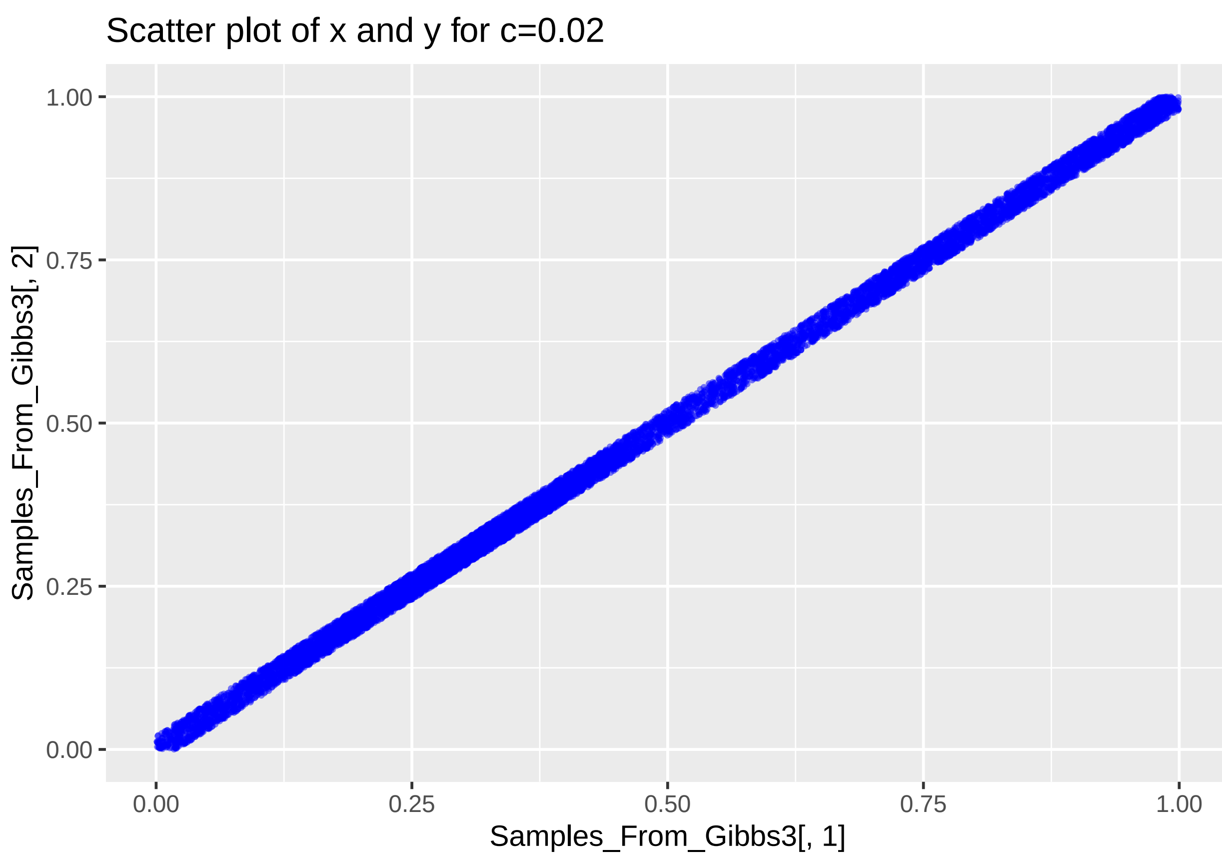
```
ts.plot(Samples_From_Gibbs3[,1], type="l", col="black", lwd=1, xlab="Iteration",
    ylab="x", main="Trace plot of x for c=0.02")
```




```
ts.plot(Samples_From_Gibbs3[,2], type="l", col="black", lwd=1, xlab="Iteration",
    ylab="y", main="Trace plot of y for c=0.02")
```




```
ggplot(data.frame(Samples_From_Gibbs3), aes(x = Samples_From_Gibbs3[,1],
    y = Samples_From_Gibbs3[,2])) + geom_point(alpha = 0.5,size = 0.5,color = "blue") + ggtitle("Scatte
```



```
par(mfrow=c(3,1))
```

(d)

As c decreases, the trace plots of x and y become more correlated. The correlation between x and y is approximately 1 as c approaches 0. It violates the independence assumption of the Gibbs sampler. So we can see that when $c = 0.02$, the marginal convergence rate is slow for both x and y.

(e)

Gibbs sampler for f(x,y)

Given

$$U = \frac{X+Y}{2}, V = \frac{X-Y}{2}$$

Thus,

$$X = U + V, Y = U - V$$

Jacobi matrix is:

$$J = \begin{bmatrix} \frac{\partial U}{\partial X} & \frac{\partial U}{\partial Y} \\ \frac{\partial V}{\partial X} & \frac{\partial V}{\partial Y} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Thus, the transformed pdf is:

$$f(U,V) \propto \mathbb{I}_{|2V|\leq c}\mathbb{I}_{|U+V|\leq 1}\mathbb{I}_{|U-V|\leq 1}|-2| = 2\mathbb{I}_{|V|\leq \frac{c}{2}}\mathbb{I}_{|U+V|\leq 1}\mathbb{I}_{|U-V|\leq 1}$$

Since $0 \leq |V| \leq \frac{c}{2}$, we only need to consider two cases:

1. $0 < c \leq 1$
2. $c > 1$

Case 1: $0 < c \leq 1$ The conditional pdf $f(V|U)$ is:

$$f(V|U) = \frac{f(U,V)}{f(U)} = \begin{cases} \frac{1}{\int_{-U}^{U} f(U,V)dV} = \frac{1}{2U}\mathbb{I}_{|V|\leq \frac{c}{2}}\mathbb{I}_{0\leq|U+V|\leq 1}\mathbb{I}_{0\leq|U-V|\leq 1}, & \text{if } U \leq \frac{c}{2} \\ \frac{1}{\int_{-\frac{c}{2}}^{\frac{c}{2}} f(U,V)dV} = \frac{1}{c}\mathbb{I}_{|V|\leq \frac{c}{2}}\mathbb{I}_{0\leq|U+V|\leq 1}\mathbb{I}_{0\leq|U-V|\leq 1}, & \text{if } \frac{c}{2} \leq U \leq 1 - \frac{c}{2} \\ \frac{1}{\int_{U-1}^{1-U} f(U,V)dV} = \frac{1}{2(1-U)}\mathbb{I}_{|V|\leq \frac{c}{2}}\mathbb{I}_{0\leq|U+V|\leq 1}\mathbb{I}_{0\leq|U-V|\leq 1}, & \text{if } 1 - \frac{c}{2} \leq U \leq 1 \end{cases} \tag{7}$$

The conditional pdf $f(U|V)$ is:

$$f(U|V) = \frac{f(U,V)}{f(V)} = \frac{1}{\int_{|V|}^{1-|V|} f(U,V)dU} = \frac{1}{1-2|V|}\mathbb{I}_{|V|\leq \frac{c}{2}}\mathbb{I}_{0\leq|U+V|\leq 1}\mathbb{I}_{0\leq|U-V|\leq 1} \tag{8}$$

Case 2: $c > 1$ The conditional pdf $f(V|U)$ is:

$$f(V|U) = \frac{f(U,V)}{f(U)} = \begin{cases} \frac{1}{\int_{-U}^{U} f(U,V)dV} = \frac{1}{2U}\mathbb{I}_{|V|\leq \frac{1}{2}}\mathbb{I}_{0\leq|U+V|\leq 1}\mathbb{I}_{0\leq|U-V|\leq 1}, & \text{if } U \leq \frac{1}{2} \\ \frac{1}{\int_{U-1}^{1-U} f(U,V)dV} = \frac{1}{2(1-U)}\mathbb{I}_{|V|\leq \frac{1}{2}}\mathbb{I}_{0\leq|U+V|\leq 1}\mathbb{I}_{0\leq|U-V|\leq 1}, & \text{if } \frac{1}{2} \leq U \leq 1 \end{cases} \tag{9}$$

The conditional pdf $f(U|V)$ is:

$$f(U|V) = \frac{f(U,V)}{f(V)} = \frac{1}{1-2|V|}\mathbb{I}_{|V|\leq\frac{1}{2}}\mathbb{I}_{0\leq|U+V|\leq1}\mathbb{I}_{0\leq|U-V|\leq1} \tag{10}$$

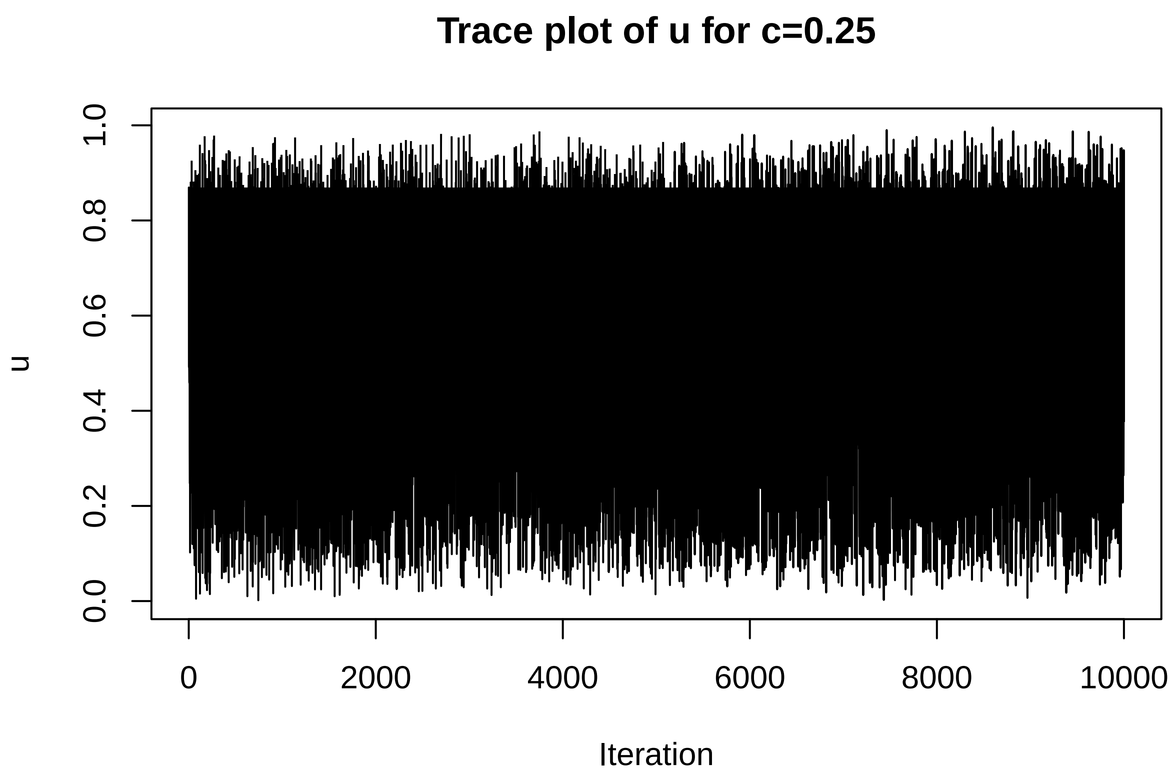
Gibbs Sampler Implementation

Gibbs sampler for (U,V) is implemented as follows:

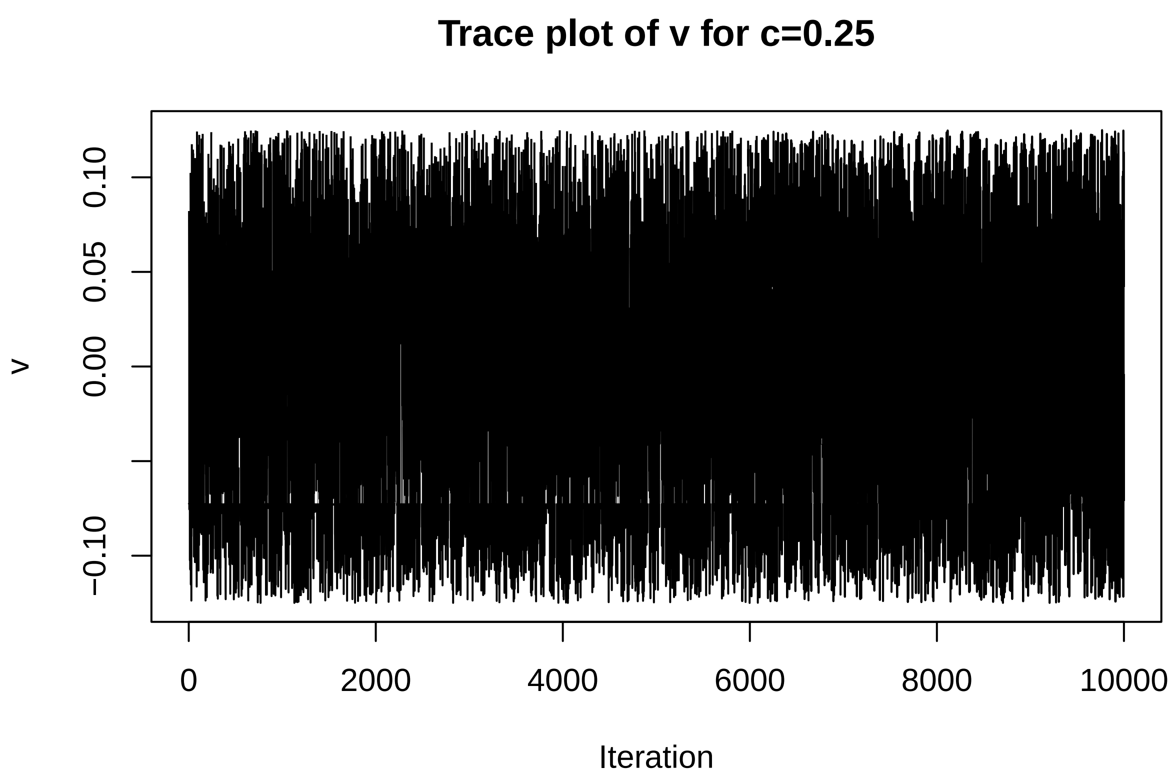
```
set.seed(20230225)

v_given_u <- function(u, c){
    if(u <= c/2){
        return(runif(1, -u, u))
    }else if(u <= 1-c/2){
        return(runif(1, -c/2, c/2))
    }else{
        return(runif(1, u-1, 1-u))
    }
}

u_given_v <- function(v, c){
    return(runif(1, abs(v), 1-abs(v)))
}

sampleGibbs2 <- function(start.u, start.v,c, n.sims,burnin=0){

    # initialize the chain
    chain <- matrix(NA, nrow=n.sims, ncol=2)
    chain[1,] <- c(start.u, start.v)

    # loop through the chain
    for(i in 2:n.sims){
        chain[i,1] <- u_given_v(chain[i-1,2], c)
        chain[i,2] <- v_given_u(chain[i,1], c)
    }

    # return the chain
    return(chain[(burnin+1):n.sims,])
}

Samples_From_Gibbs1 = sampleGibbs2(0.5,0.5,0.25,11000,1000)
Samples_From_Gibbs2 = sampleGibbs2(0.5,0.5,0.05,11000,1000)
Samples_From_Gibbs3 = sampleGibbs2(0.5,0.5,0.02,11000,1000)
```

Traceplots and Scatterplots

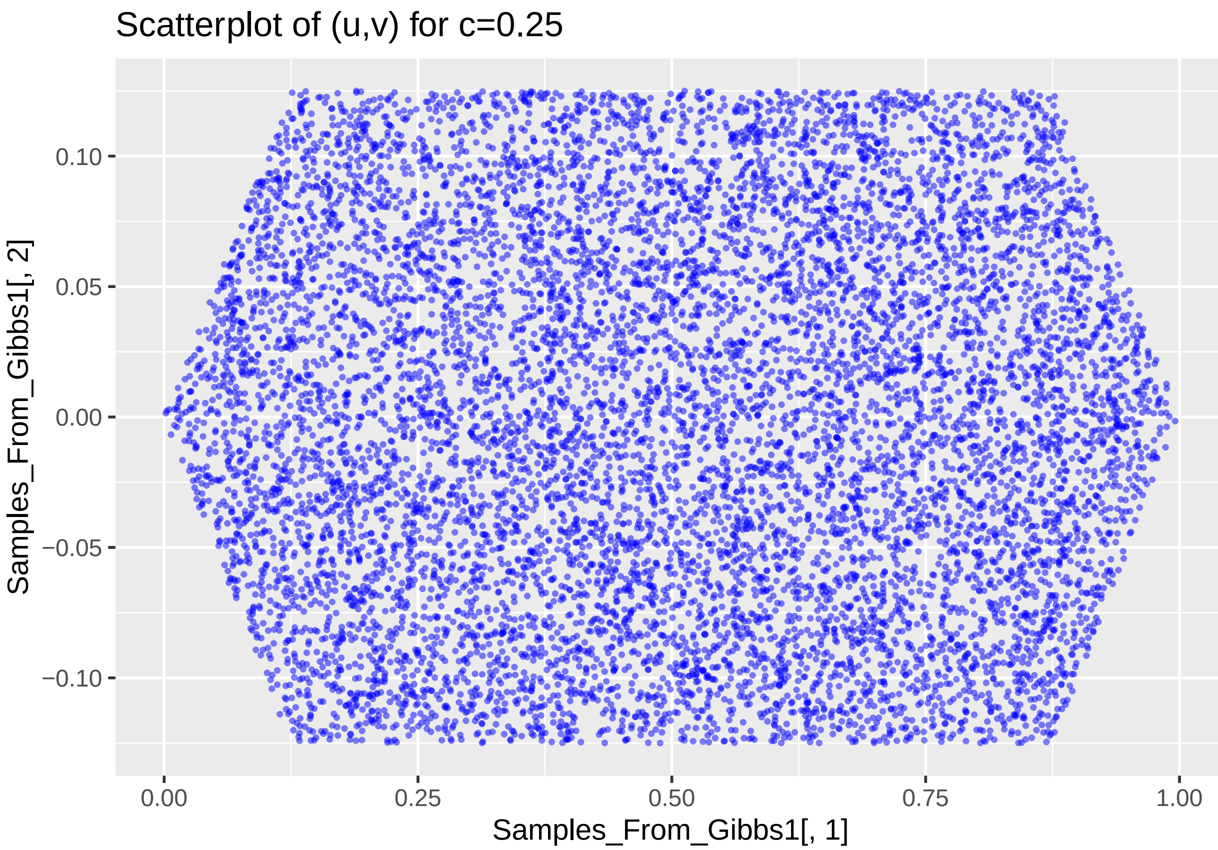
```
ts.plot(Samples_From_Gibbs1[,1], type="l", col="black", lwd=1, xlab="Iteration",
 ylab="u", main="Trace plot of u for c=0.25")
```




```r
ts.plot(Samples_From_Gibbs1[,2], type="l", col="black", lwd=1, xlab="Iteration",
ylab="v", main="Trace plot of v for c=0.25")
```

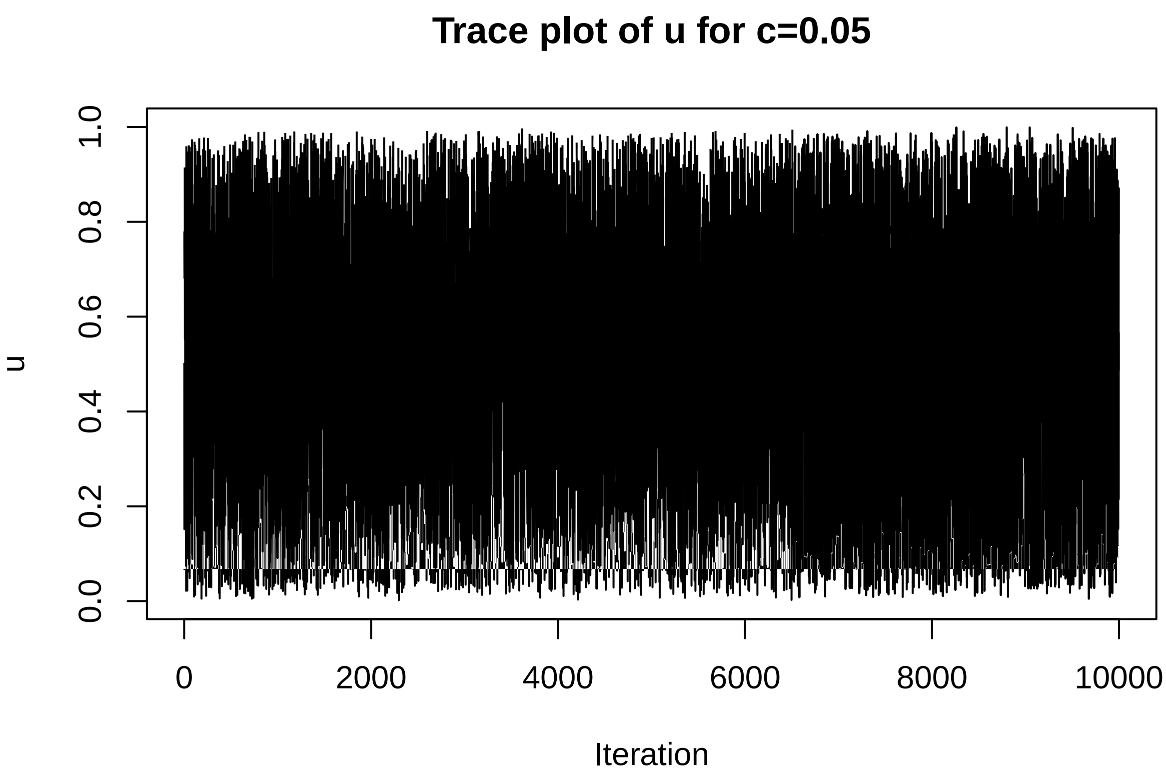



```r
ggplot(data.frame(Samples_From_Gibbs1), aes(x=Samples_From_Gibbs1[,1], y=Samples_From_Gibbs1[,2])) + ge
```

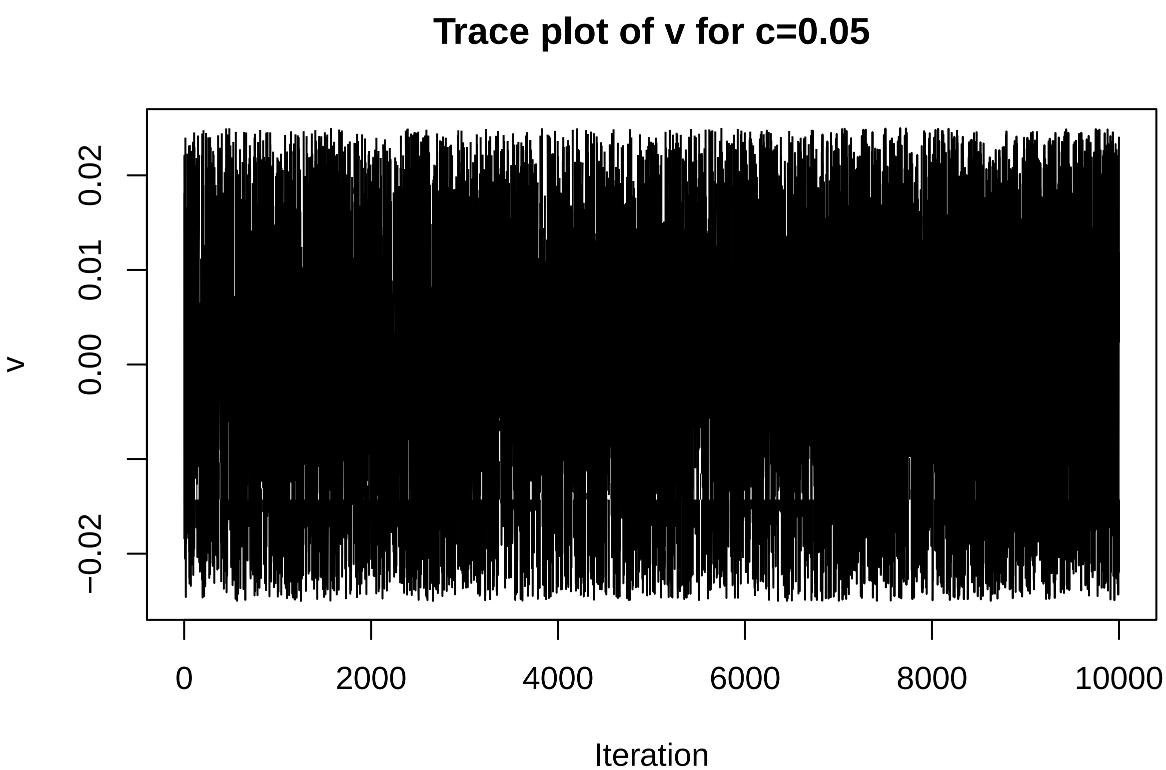



```r
par(mfrow=c(3,1))
```

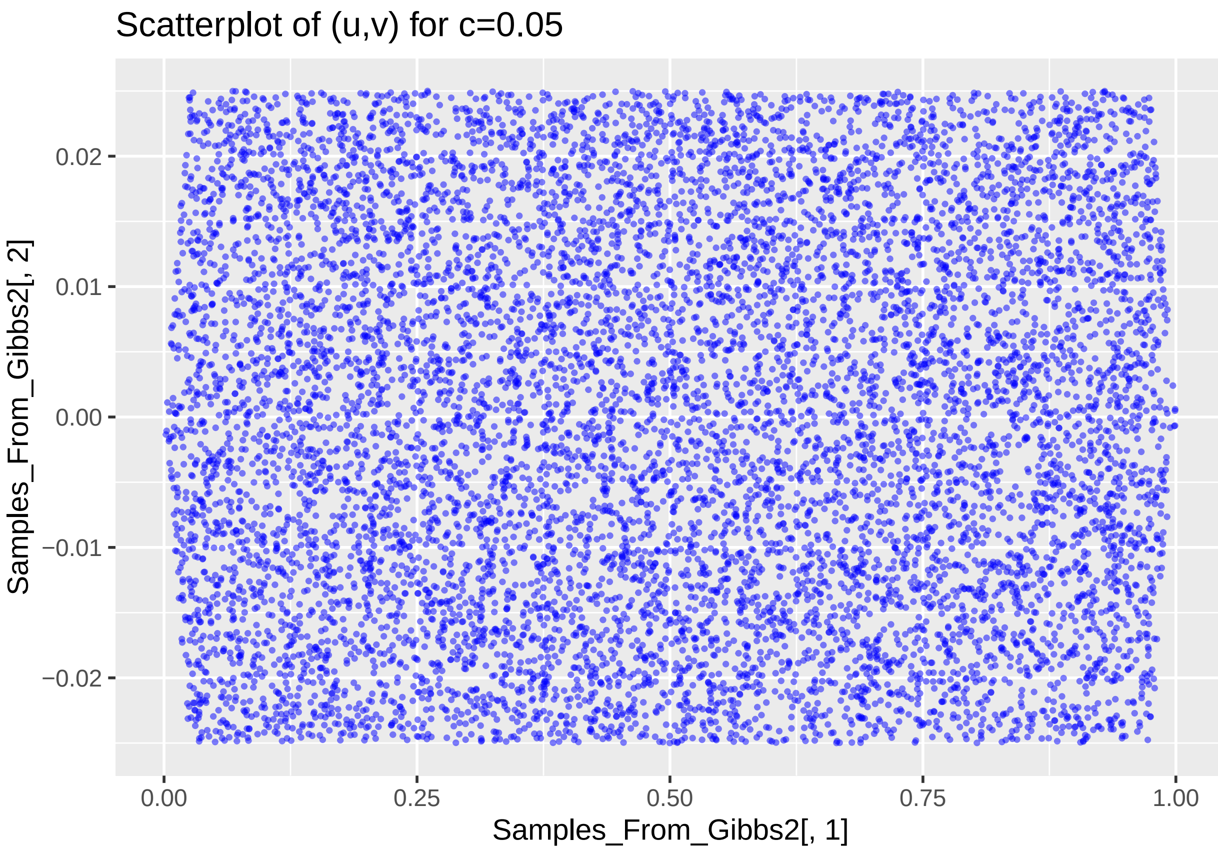
```r
ts.plot(Samples_From_Gibbs2[,1], type="l", col="black", lwd=1, xlab="Iteration",
 ylab="u", main="Trace plot of u for c=0.05")
```




```
ts.plot(Samples_From_Gibbs2[,2], type="l", col="black", lwd=1, xlab="Iteration",
ylab="v", main="Trace plot of v for c=0.05")
```

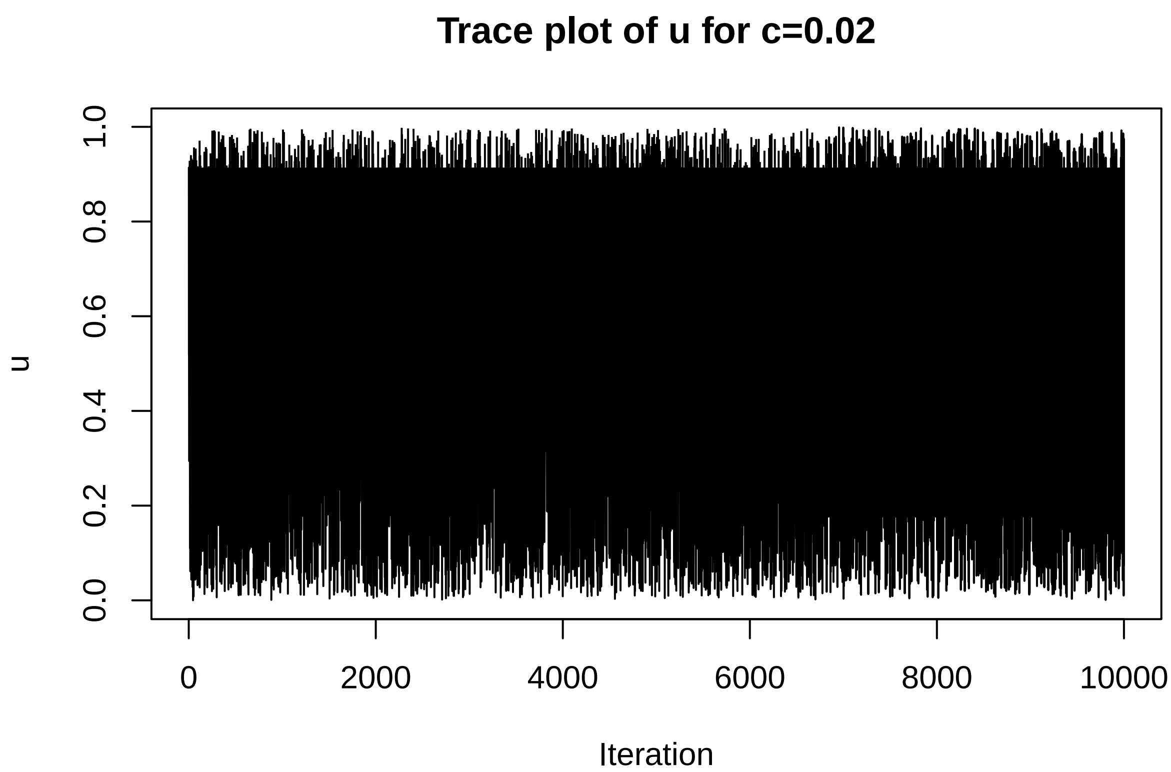



```
ggplot(data.frame(Samples_From_Gibbs2), aes(x=Samples_From_Gibbs2[,1], y=Samples_From_Gibbs2[,2])) + ge
```

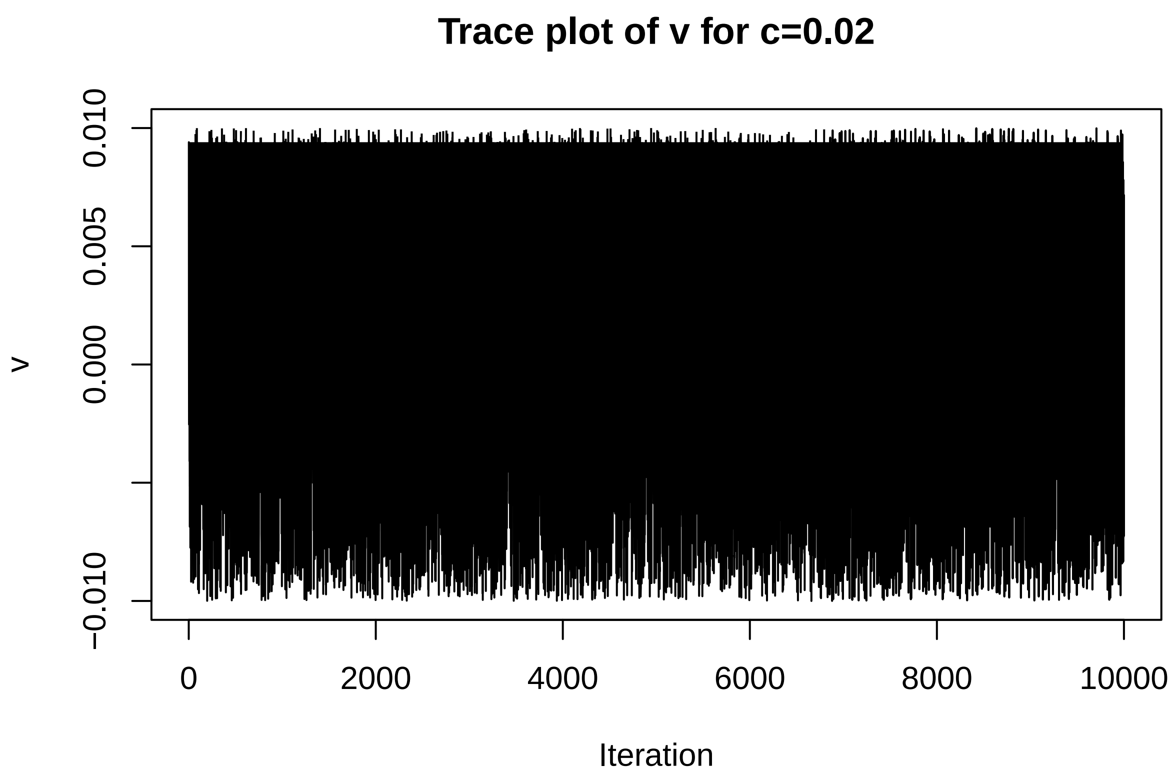



```
par(mfrow=c(3,1))
```

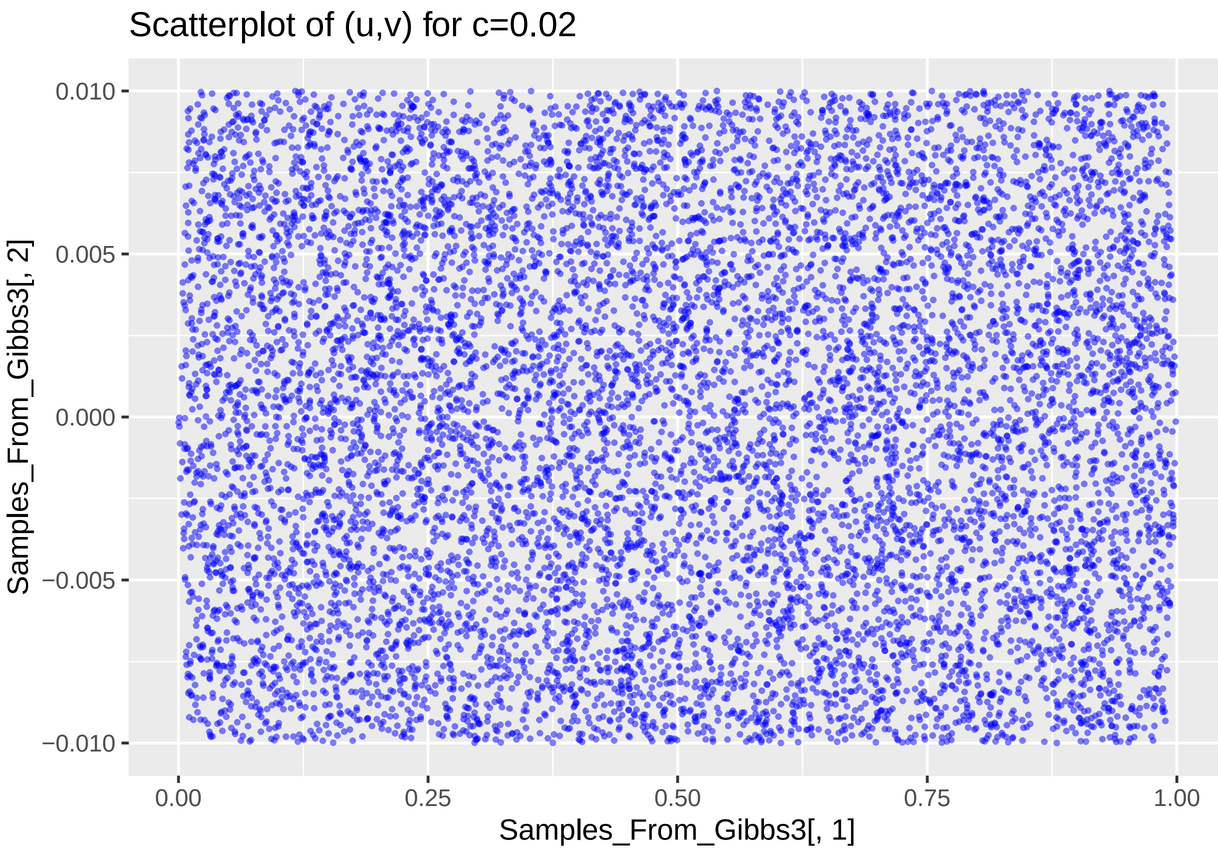
```
ts.plot(Samples_From_Gibbs3[,1], type="l", col="black", lwd=1, xlab="Iteration",
 ylab="u", main="Trace plot of u for c=0.02")
```




```
ts.plot(Samples_From_Gibbs3[,2], type="l", col="black", lwd=1, xlab="Iteration",
ylab="v", main="Trace plot of v for c=0.02")
```




```
ggplot(data.frame(Samples_From_Gibbs3), aes(x=Samples_From_Gibbs3[,1], y=Samples_From_Gibbs3[,2])) + ge
```




```
par(mfrow=c(3,1))
```

Metropolis-Hastings Sampler Implementation

Metropolis-Hastings sampler for (X,Y) is implemented as follows:

```
density <- function(x,y,c){
    if(abs(x-y) <= c & x>=0 & x<=1 & y>=0 & y<=1){
        return(1)
    }else{
        return(0)
    }
}
```

```
# density function for mixed normal distribution
density <- function(x,y,c){
    if(abs(x-y) <= c & x>=0 & x<=1 & y>=0 & y<=1){
        return(1)
    }else{
        return(0)
    }
}
```

```
set.seed(20230225)
sampleL <- function(start.a, start.b,c, n.sims,burnin=0){}
```

```
set.seed(20230225)
```

```r
# Define the mixture parameters
p1 <- 1/5
p2 <- 4/5
mu1 <- c(-5, -5)
mu2 <- c(5, 5)
sigma <- diag(2)

# Define the PDF function
pdf_mixture <- function(x, p1, p2, mu1, mu2, sigma) {
  dens1 <- p1 * dmvnorm(x, mean = mu1, sigma = sigma)
  dens2 <- p2 * dmvnorm(x, mean = mu2, sigma = sigma)
  return(dens1 + dens2)
}

gradient_mixture <- function(x, p1, p2, mu1, mu2, sigma){
  dens1 <- p1 * dmvnorm(x, mean = mu1, sigma = sigma)
  dens2 <- p2 * dmvnorm(x, mean = mu2, sigma = sigma)
  numer <- dens1 + dens2
  grad_x1 <- (p1 * dmvnorm(x, mean = mu1, sigma = sigma) / numer) *(-0.5) * solve(sigma) %*% (x - mu1)
  grad_x2 <- (p2 * dmvnorm(x, mean = mu2, sigma = sigma) / numer) *(-0.5) * solve(sigma) %*% (x - mu2)
return(grad_x1+grad_x2)
}

pdf_mixture(c(5,5), p1, p2, mu1, mu2, sigma)
```

```
## [1] 0.127324
```

```r
gradient_mixture(c(5,4), p1, p2, mu1, mu2, sigma)
```

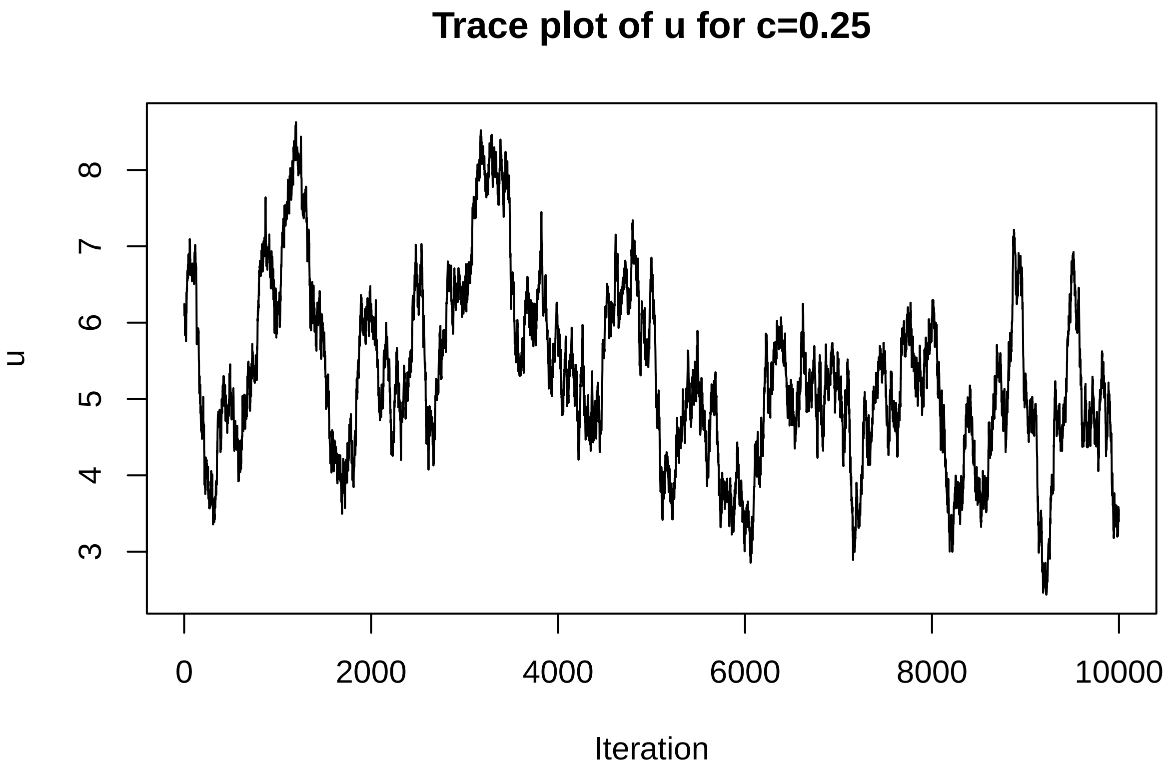
```
##                [,1]
## [1,] -1.024252e-39
## [2,]  5.000000e-01
```

```r
set.seed(20230225)
sampleMH <- function(start.a, start.b, n.sims,burnin=0){

# initialize the chain
chain <- matrix(NA, nrow=n.sims, ncol=2)
chain[1,] <- c(start.a, start.b)

# loop through the chain
for(i in 2:n.sims){
    new.a <- rnorm(2,chain[i-1],0.1)
    if(pdf_mixture(new.a,p1,p2,mu1,mu2,sigma) <= 0){
        chain[i,] <- chain[i-1,]
    } else{
        ratio <- pdf_mixture(new.a,p1,p2,mu1,mu2,sigma)/pdf_mixture(chain[i-1,],p1,p2,mu1,mu2,sigma)
    if(runif(1) < ratio){
        chain[i,] <- new.a
    }else{
        chain[i,] <- chain[i-1,]
    }
}
}
```

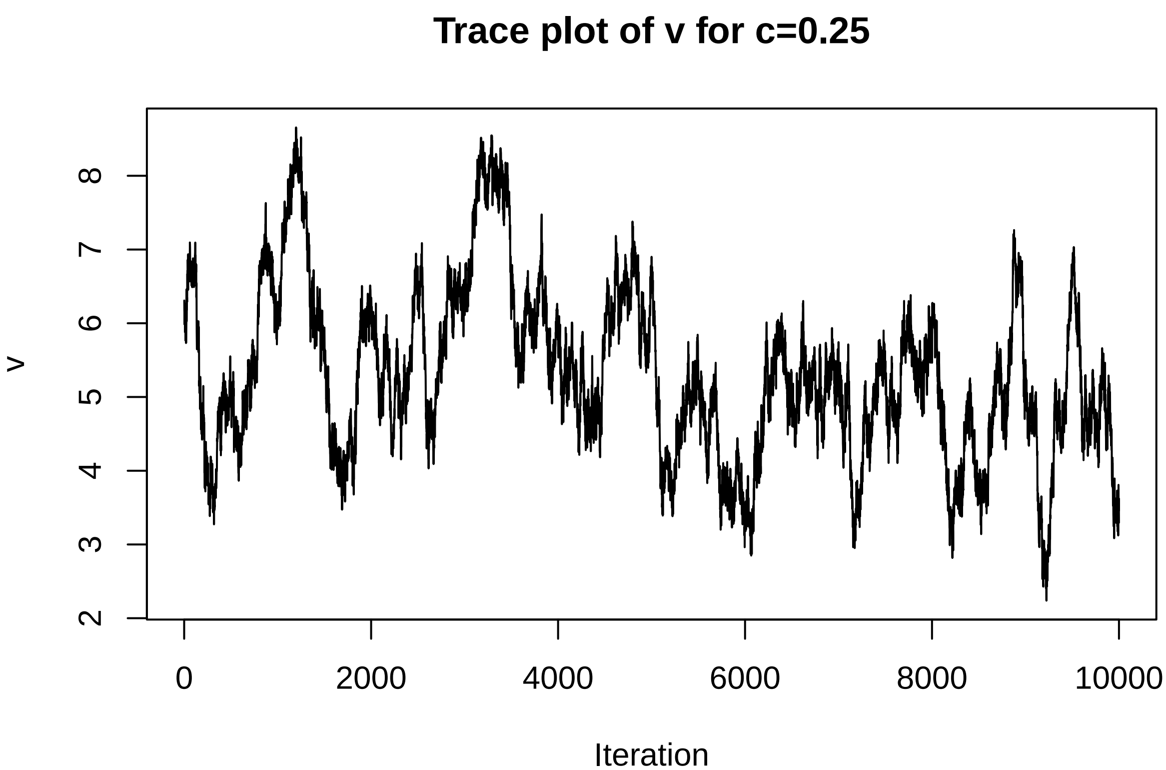
```
# return the chain
return(chain[(burnin+1):n.sims,])
}

Samples_From_MH = sampleMH(0,0,11000,1000)
```

Traceplots and Scatterplots

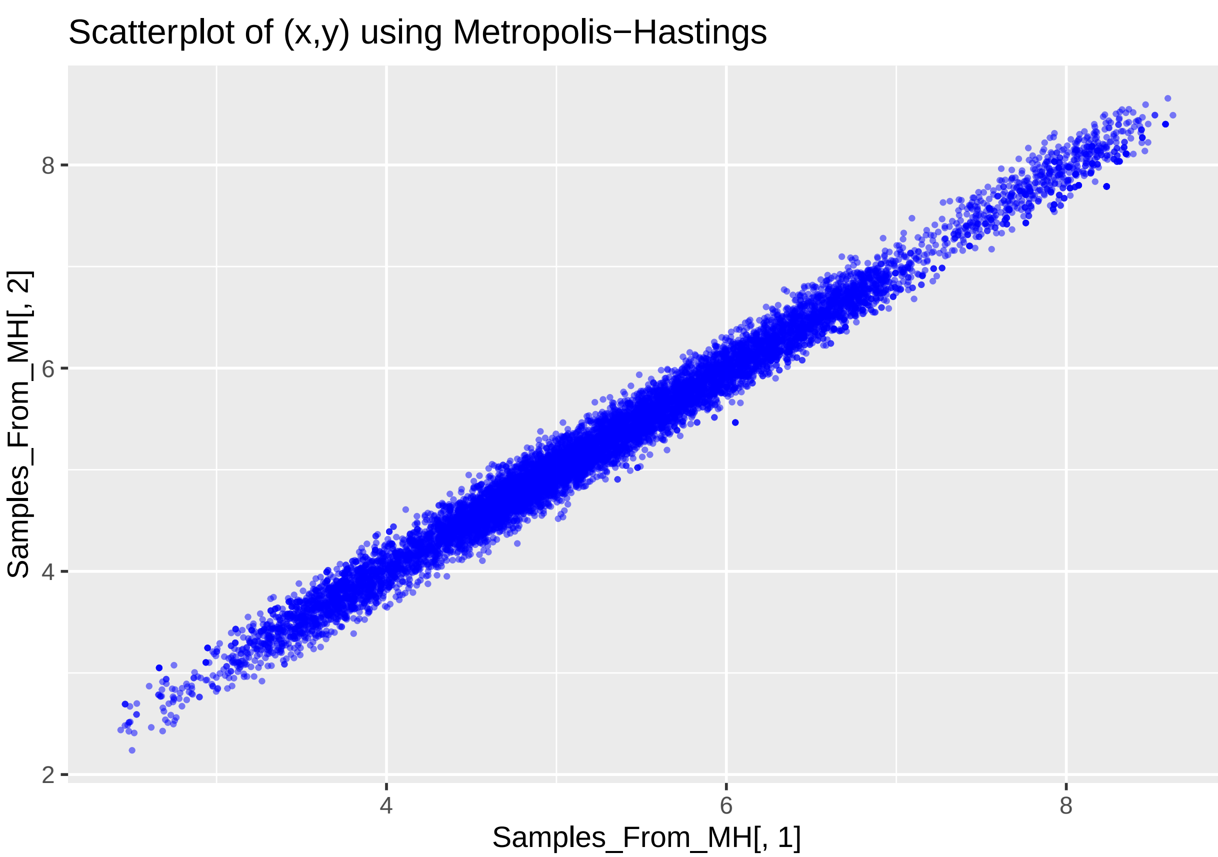
```
ts.plot(Samples_From_MH[,1], type="l", col="black", lwd=1, xlab="Iteration",
 ylab="u", main="Trace plot of u for c=0.25")
```




```
ts.plot(Samples_From_MH[,2], type="l", col="black", lwd=1, xlab="Iteration",
ylab="v", main="Trace plot of v for c=0.25")
```




```
ggplot(data.frame(Samples_From_MH), aes(x=Samples_From_MH[,1], y=Samples_From_MH[,2])) + geom_point(alp
```



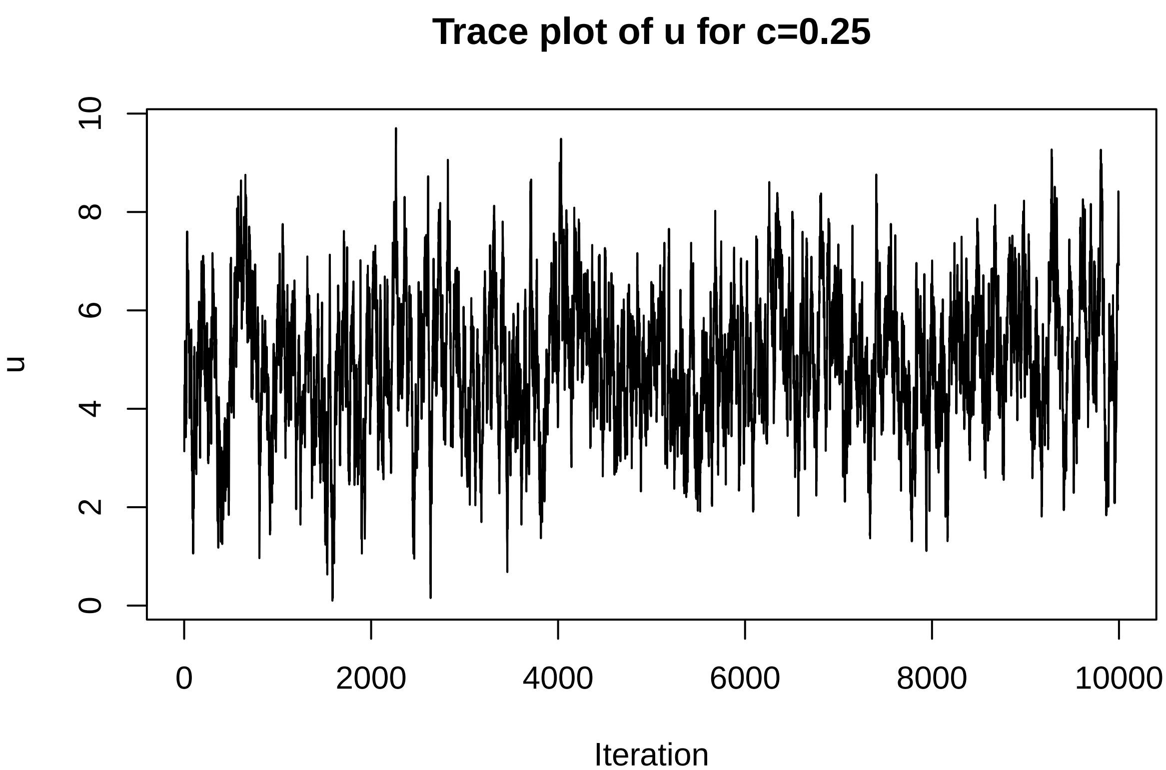
Langevin Sampler Implementation

Langevin sampler for mixture gaussian

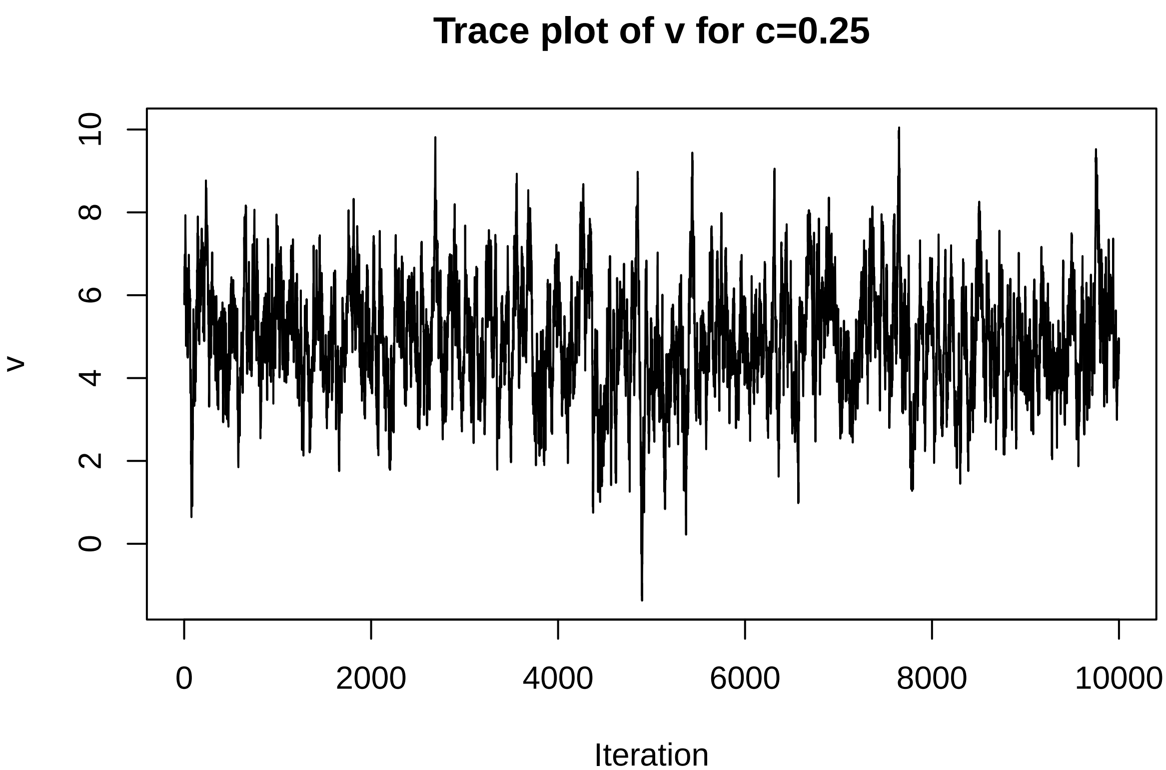
```
set.seed(20230225)
sampleL <- function(start.a, start.b, n.sims,burnin=0,stepsize=0.1){
# initialize the chain
chain <- matrix(NA, nrow=n.sims, ncol=2)
chain[1,] <- c(start.a, start.b)


# loop through the chain
for(i in 2:n.sims){
    noise <- rnorm(2, mean = 0, sd = sqrt(2 * stepsize))
    chain[i,] <- chain[i-1,] + stepsize * gradient_mixture(chain[i-1,],p1,p2,mu1,mu2,sigma) + noise
}
# return the chain
return(chain[(burnin+1):n.sims,])
}

Samples_From_Langevin = sampleL(0,0,11000,1000,0.1)
```

Traceplots and Scatterplots

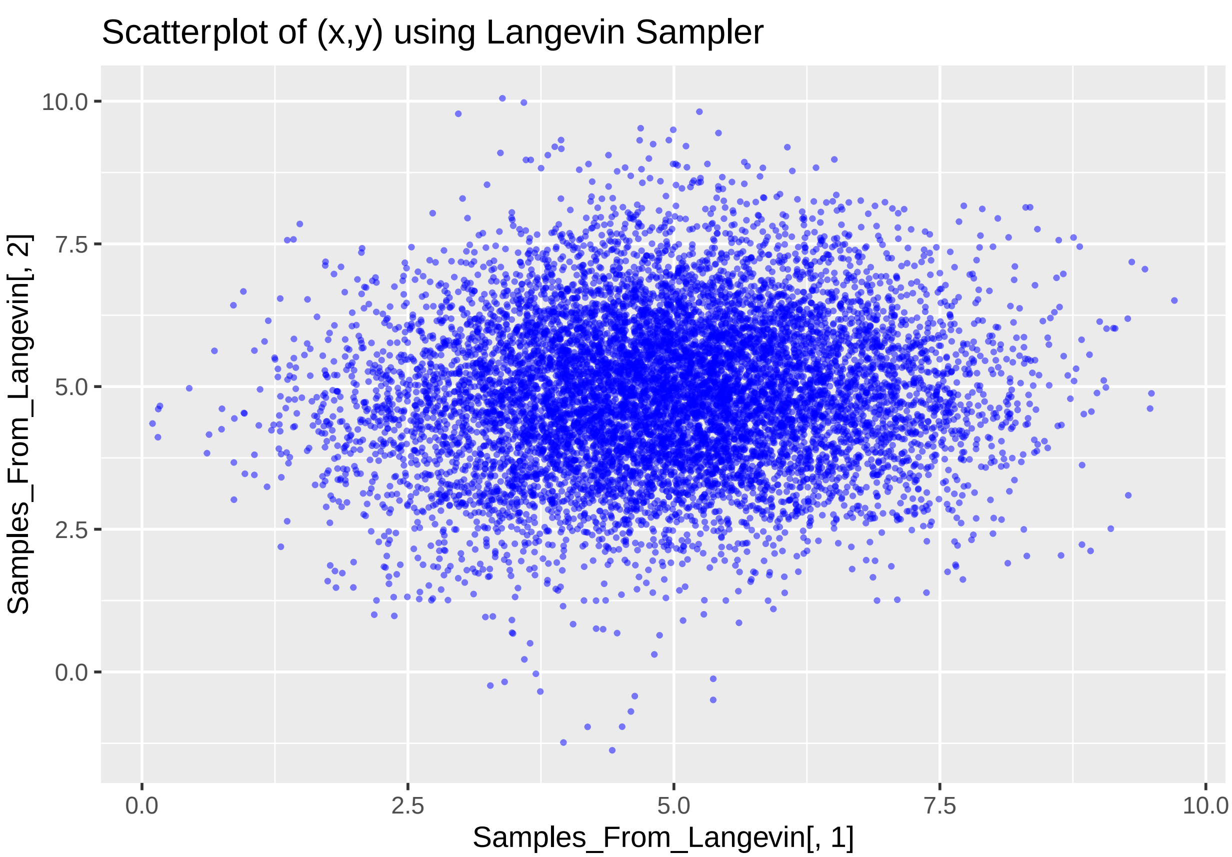
```
ts.plot(Samples_From_Langevin[,1], type="l", col="black", lwd=1, xlab="Iteration",
 ylab="u", main="Trace plot of u for c=0.25")
```




```
ts.plot(Samples_From_Langevin[,2], type="l", col="black", lwd=1, xlab="Iteration",
ylab="v", main="Trace plot of v for c=0.25")
```




```
ggplot(data.frame(Samples_From_Langevin), aes(x=Samples_From_Langevin[,1], y=Samples_From_Langevin[,2]))
```

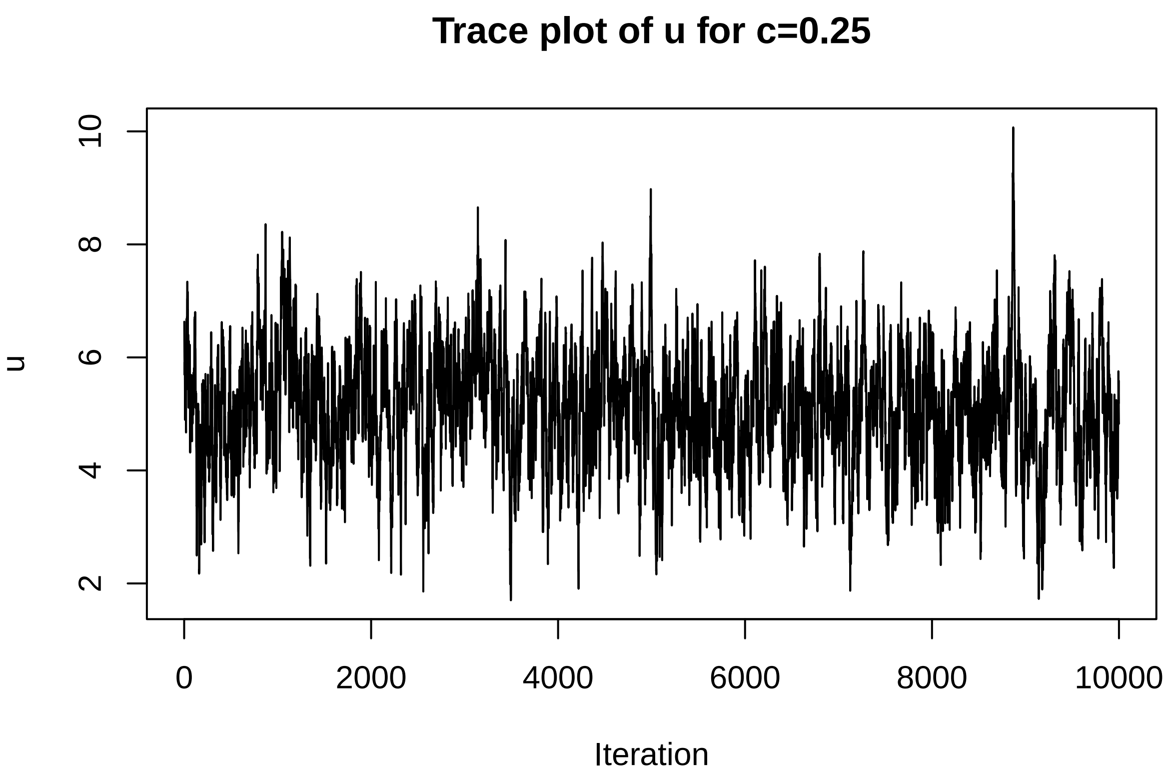


Metropolis-adjusted Langevin algorithm Implementation

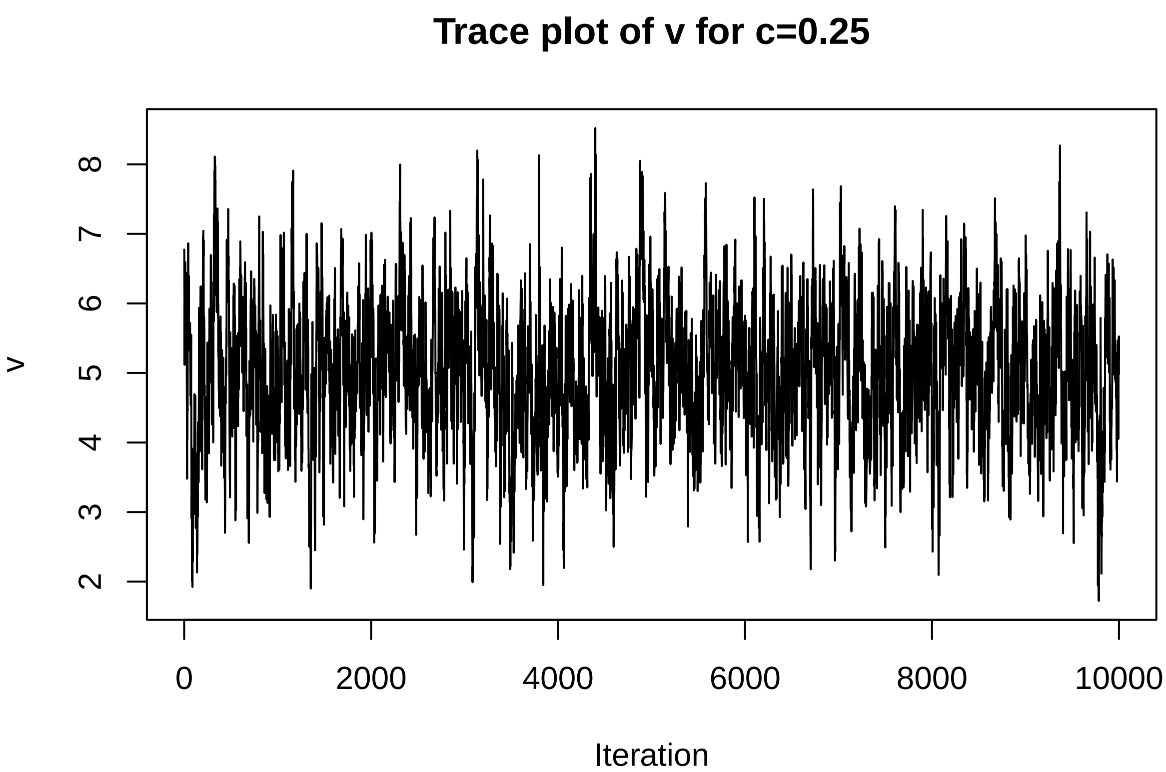
```
set.seed(20230225)

q <- function(x,y,p1,p2,mu1,mu2,sigma,stepsize){
    return (exp(-1/(4*stepsize)*sum((x-y-stepsize*gradient_mixture(y,p1,p2,mu1,mu2,sigma))**2)))
}






sampleMALA <- function(start.a, start.b, n.sims,burnin=0,stepsize=0.1){
# initialize the chain
chain <- matrix(NA, nrow=n.sims, ncol=2)
chain[1,] <- c(start.a, start.b)


# loop through the chain
for(i in 2:n.sims){
    noise <- rnorm(2, mean = 0, sd = sqrt(2 * stepsize))
    x_new <-  as.vector(chain[i-1,] + stepsize * gradient_mixture(chain[i-1,],p1,p2,mu1,mu2,sigma) + noi
    ratio <- pdf_mixture(x_new,p1,p2,mu1,mu2,sigma)*q(chain[i-1,],x_new,p1,p2,mu1,mu2,sigma,stepsize)/(p
    if(runif(1) < ratio){
        chain[i,] <- x_new
}else{
    chain[i,] <- chain[i-1,]
}
}
# return the chain
return(chain[(burnin+1):n.sims,])
}
Samples_From_MALA = sampleMALA(0,0,11000,1000,0.1)
```

Traceplots and Scatterplots

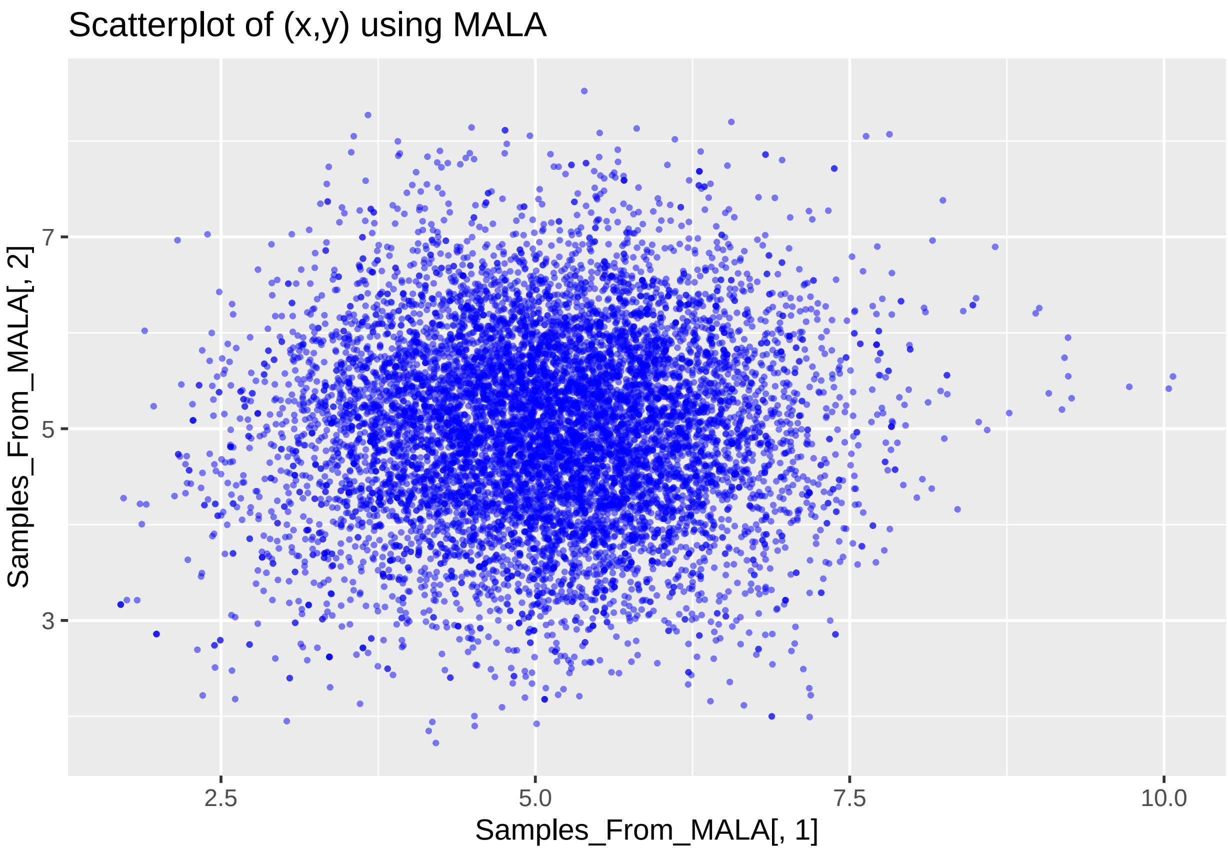
```
ts.plot(Samples_From_MALA[,1], type="l", col="black", lwd=1, xlab="Iteration",
 ylab="u", main="Trace plot of u for c=0.25")
```




```
ts.plot(Samples_From_MALA[,2], type="l", col="black", lwd=1, xlab="Iteration",
ylab="v", main="Trace plot of v for c=0.25")
```




```
ggplot(data.frame(Samples_From_MALA), aes(x=Samples_From_MALA[,1], y=Samples_From_MALA[,2])) + geom_poi
```



‘‘‘

Hamiltonian Monte Carlo Implementation