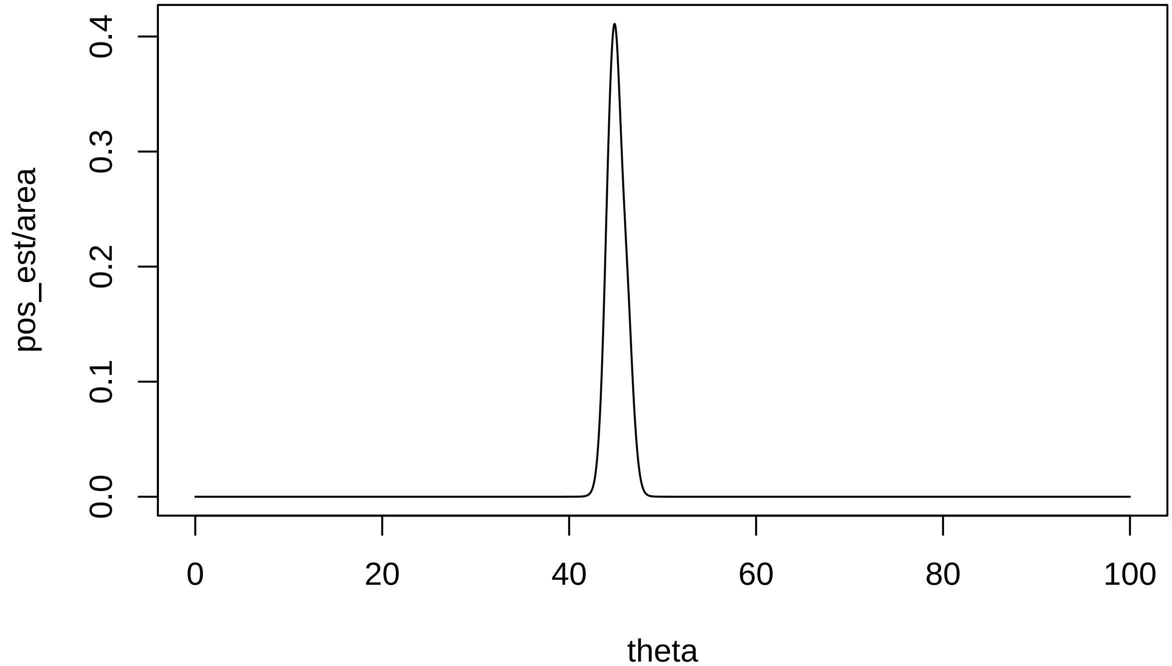
hw1

Qi Wang

2022/3/29

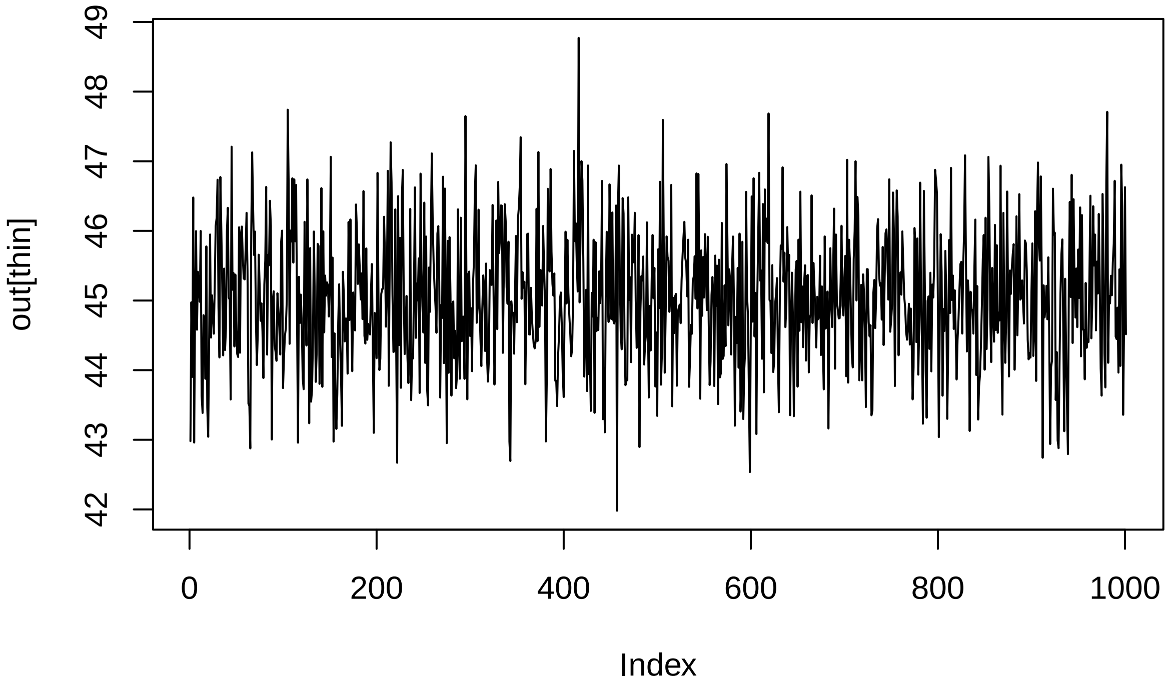
Question 3.a: Cauchy Distribution Simulation

```
l <- function(x, theta){
  out <- 1 / (1+ (x-theta)^2)
  return(out)
}

theta <- seq(from = 0, to = 100, by = 0.01)

y <- c(43, 44, 45, 46.5, 47.5)
pos_est <-rep(NA, length(theta))

for (i in 1:length(theta)) {

  pos_est[i] <- 1/100 *  prod(l(x = y, theta = theta[i]))

}
area <- sum(pos_est*0.01)

plot(x = theta, y = pos_est/area, type = 'l')
```



Question 3.b: MH to get the Samples

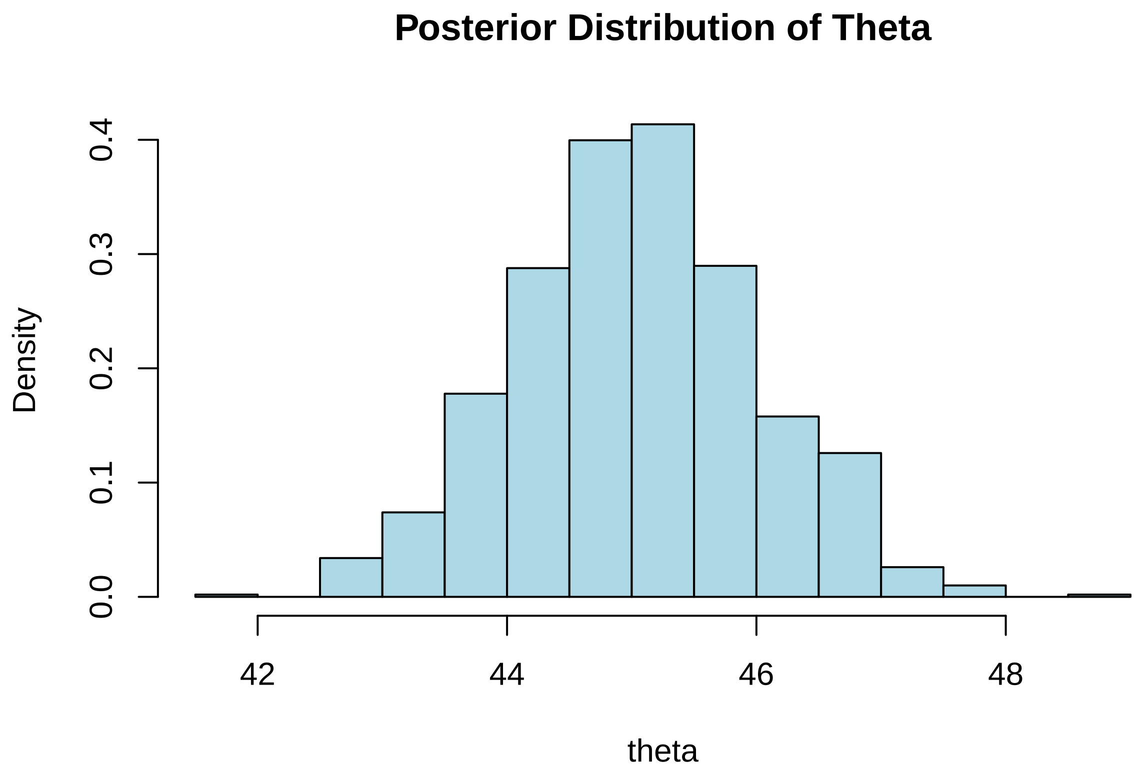
```
out <- rep(NA, 10000)
out[1] <- 41
i <- 1
for (i in 1:9999) {
  new <- rnorm(1, out[i], sd = 2)
  p <- prod(l(x = y, theta = new))/prod(l(x = y, theta = out[i]))
  index <- runif(1)
  if(index <= p){
    out[i+1] <- new
  }else{
    out[i+1] <- out[i]
  }
}
thin <- seq(from = 2000, to = 10000, by = 8)
plot(out[thin], type = 'l')
```




```
mcmcse::ess(out[thin])
```

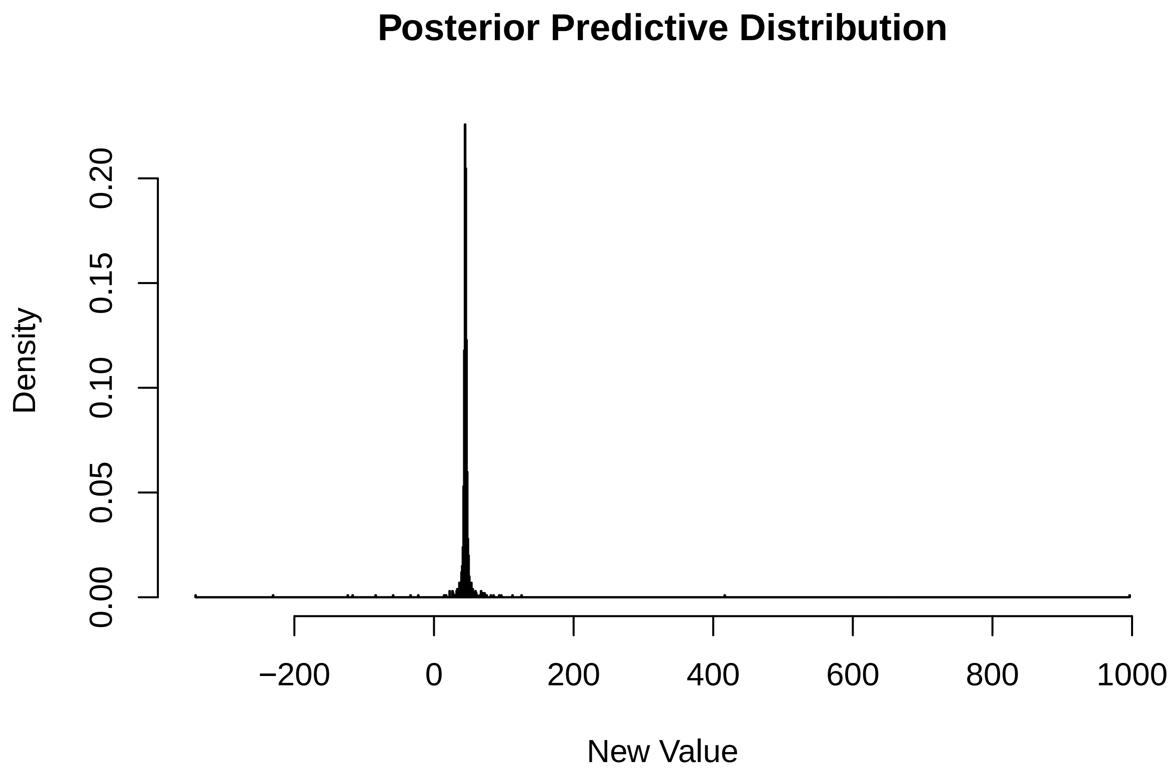
```
## [1] 960.7548
```

```
hist(out[thin], xlab = "theta", probability = TRUE, main = "Posterior Distribution of Theta", col = "lig
```



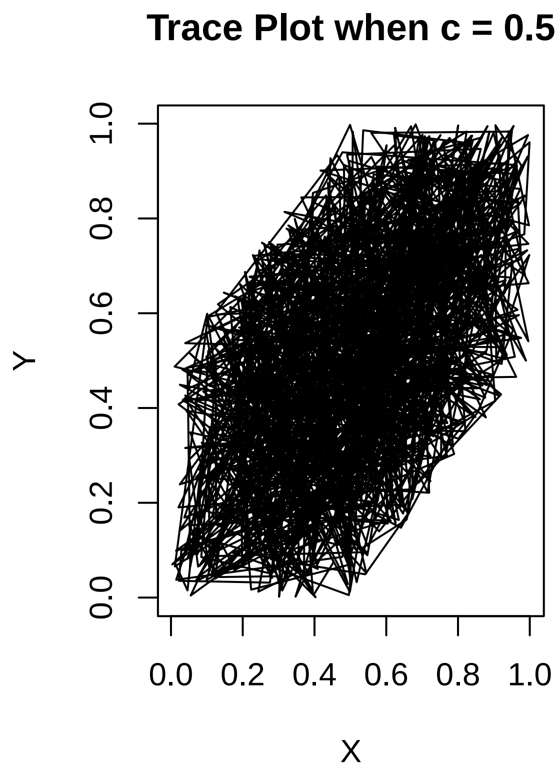
Question 3.c: Generate Posterior Predictive Distribution

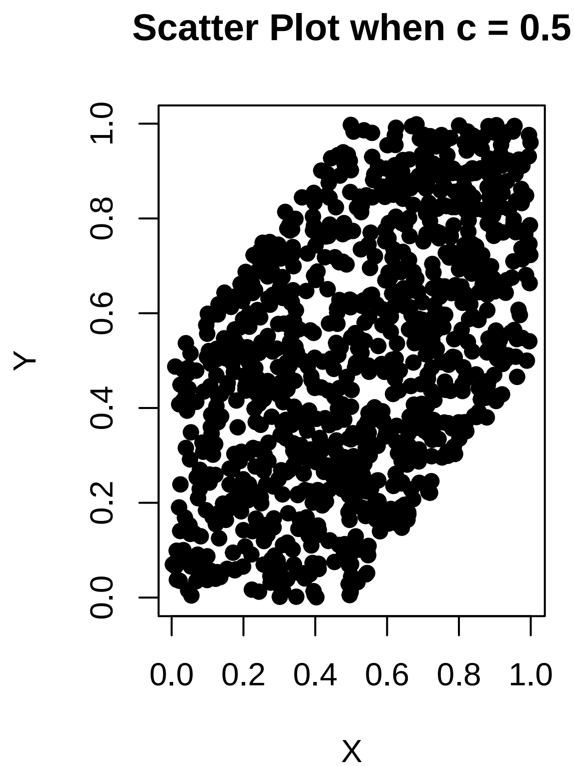
```
set.seed(2)
theta_pos <- out[thin]
i <- 1
pred <- rep(NA, length(theta_pos))
for (i in 1:length(theta_pos)) {
  pred[i] <- rcauchy(1, location = theta_pos[i], scale = 1)
}
hist(pred, main = "Posterior Predictive Distribution", xlab = "New Value", probability  = TRUE, col = "l
```



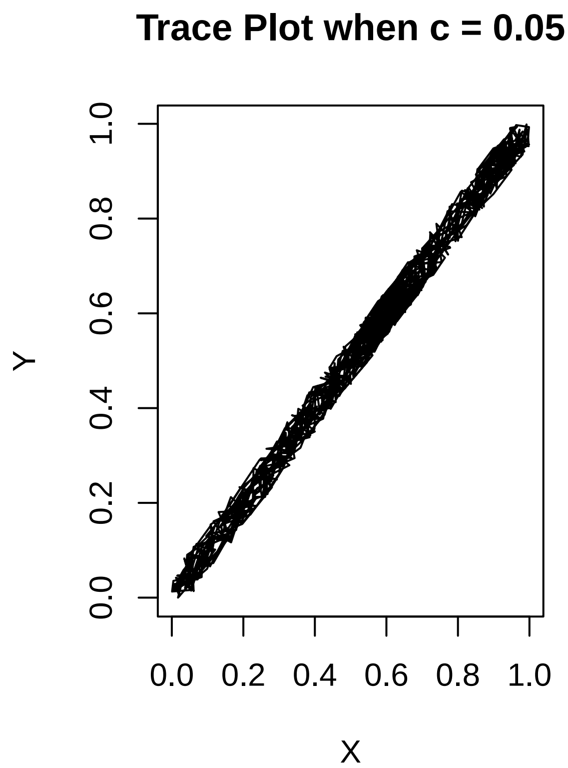
Question 4: Gibbs Sampler for 2-d Uniform Distribution

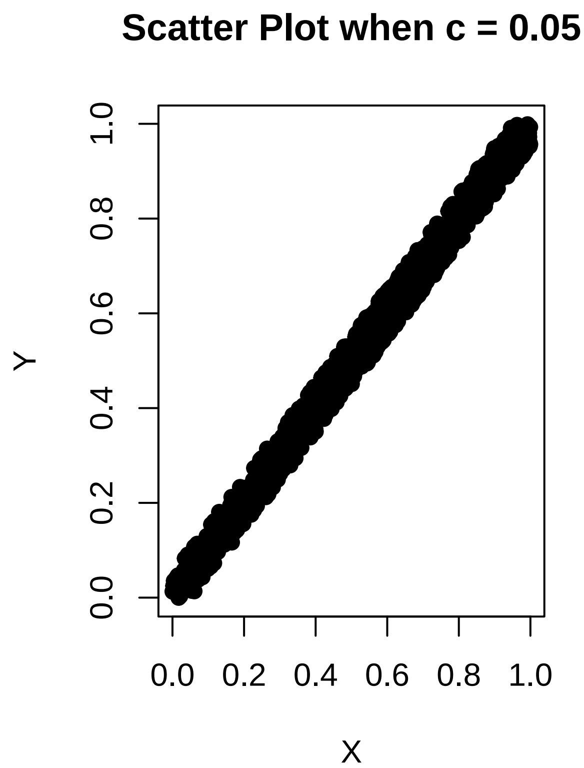
```
par(mfrow = c(1,2))
c <- 0.5
gibbs <- matrix(NA, nrow = 2, ncol = 1000)
gibbs[1,1] <- 0.5

for (i in 1:999) {
  L_1 <- max(0, gibbs[1,i] - c)
  U_1 <- min(1, gibbs[1,i] + c)
  gibbs[2,i] <- runif(1, L_1, U_1)
  L_2 <- max(0, gibbs[2,i] - c)
  U_2 <- min(1, gibbs[2,i] + c)
  gibbs[1,i+1] <- runif(1, L_2, U_2)
}
plot(x = gibbs[1,], y = gibbs[2,], type = 'l', xlab = "X", ylab = "Y", main = "Trace Plot when c = 0.5")
plot(x = gibbs[1,], y = gibbs[2,], type = 'p', pch = 19, xlab = "X", ylab = "Y", main = "Scatter Plot wh
```



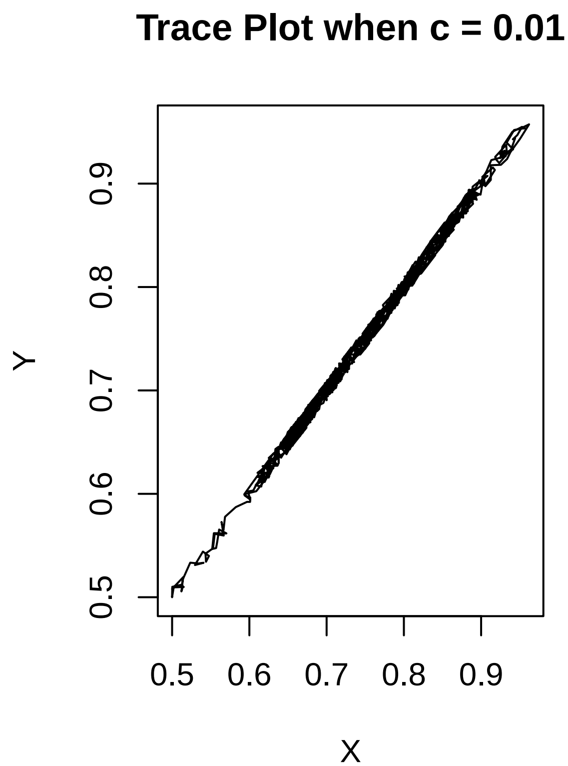


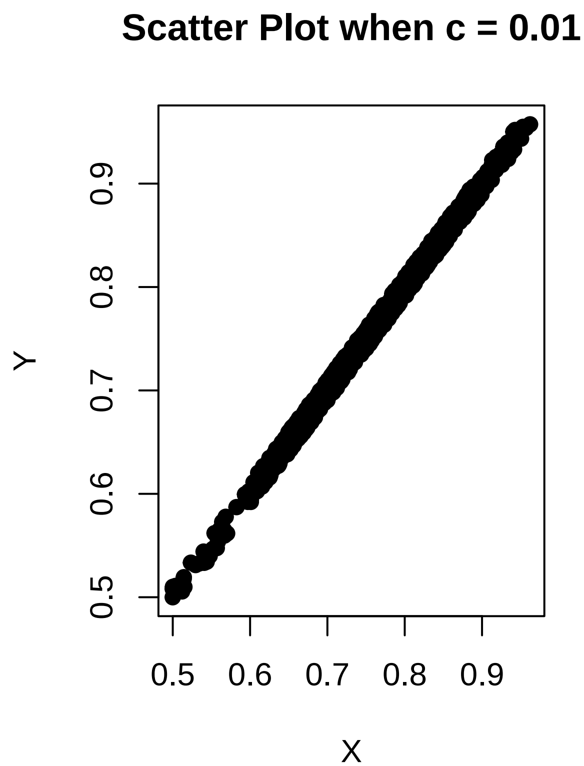

```
par(mfrow = c(1,2))
c <- 0.05
gibbs <- matrix(NA, nrow = 2, ncol = 1000)
gibbs[1,1] <- 0.5

for (i in 1:999) {
  L_1 <- max(0, gibbs[1,i] - c)
  U_1 <- min(1, gibbs[1,i] + c)
  gibbs[2,i] <- runif(1, L_1, U_1)
  L_2 <- max(0, gibbs[2,i] - c)
  U_2 <- min(1, gibbs[2,i] + c)
  gibbs[1,i+1] <- runif(1, L_2, U_2)
}
plot(x = gibbs[1,], y = gibbs[2,], type = 'l', xlab = "X", ylab = "Y", main = "Trace Plot when c = 0.05
plot(x = gibbs[1,], y = gibbs[2,], type = 'p', pch = 19, xlab = "X", ylab = "Y", main = "Scatter Plot wh
```





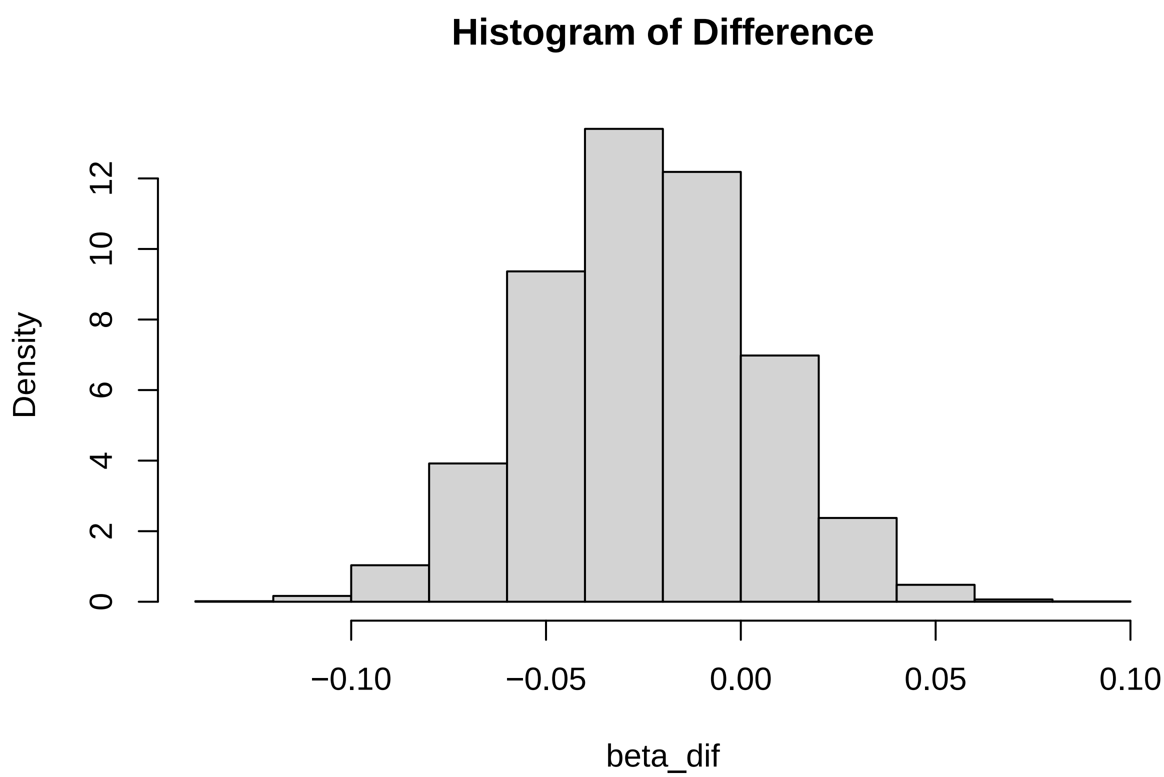

```
par(mfrow = c(1,2))
c <- 0.01
gibbs <- matrix(NA, nrow = 2, ncol = 1000)
gibbs[1,1] <- 0.5

for (i in 1:999) {
  L_1 <- max(0, gibbs[1,i] - c)
  U_1 <- min(1, gibbs[1,i] + c)
  gibbs[2,i] <- runif(1, L_1, U_1)
  L_2 <- max(0, gibbs[2,i] - c)
  U_2 <- min(1, gibbs[2,i] + c)
  gibbs[1,i+1] <- runif(1, L_2, U_2)
}
plot(x = gibbs[1,], y = gibbs[2,], type = 'l', xlab = "X", ylab = "Y", main = "Trace Plot when c = 0.01
plot(x = gibbs[1,], y = gibbs[2,], type = 'p', pch = 19, xlab = "X", ylab = "Y", main = "Scatter Plot wh
```





Question 6: Difference Between Beta Distribution

```
beta_1 <- rbeta(10000, shape1 = 296, shape2 = 309)
beta_2 <- rbeta(10000, shape1 = 290, shape2 = 334)
beta_dif <- beta_2 - beta_1
hist(beta_dif, probability = TRUE, main = "Histogram of Difference")
```



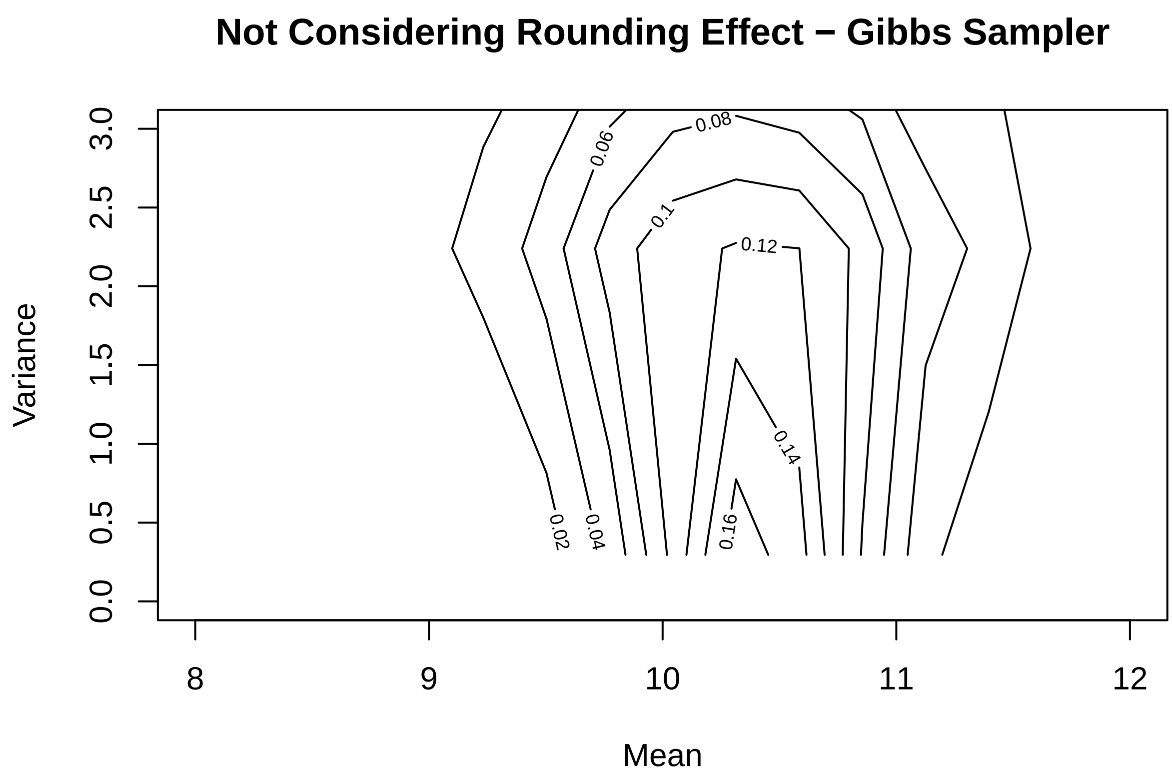

```r
length(which(beta_dif>=0))/length(beta_dif)
```

```
## [1] 0.1982
```

Question 7:

Not considering the Rounding Effect _ Method 1 - Gibbs Sampler

```r
set.seed(0)
mu = sig <- rep(NA, 10000)
mu[1] <- 10
y <- c(10, 10, 11, 12, 9)
i <- 1

for (i in 1:10000) {
  sig_inv <- rgamma(1, shape = 5/2, rate = sum( (y-mu[i])^2)/2 )
  sig[i] <- 1/sig_inv
  mu[i+1] <- rnorm(1, mean = mean(y), sd = sqrt(1/5 * sig[i]))
}
thin <- seq(from = 2000, to = 10000, by = 8)
mu <- mu[thin]
sig <- sig[thin]
den <- MASS::kde2d(x = mu, y = sig)
contour(x = den$x, y = den$y, z = den$z, xlim = c(8, 12), ylim = c(0, 3),
        xlab = "Mean", ylab = "Variance", main = "Not Considering Rounding Effect - Gibbs Sampler")
```




```
sum_1 <- matrix( round(
  c(quantile(mu, 0.025), mean(mu) ,quantile(mu, 0.975), var(mu),
    quantile(sig, 0.025), mean(sig) ,quantile(sig, 0.975), var(sig)),2 ) , nrow = 2,
  ncol = 4, byrow = TRUE
)
colnames(sum_1) <- c("2.5% Lower", "Mean", "97.5% Upper", "Variance")
rownames(sum_1) <- c("mu", "sigma")
knitr::kable(sum_1, align = 'c', caption = "Not Considering Rounding Effect - Gibbs Sampler")
```

Table 1: Not Considering Rounding Effect - Gibbs Sampler

	2.5% Lower	Mean	97.5% Upper	Variance
mu	8.91	10.36	11.76	0.47
sigma	0.48	2.46	10.49	10.72

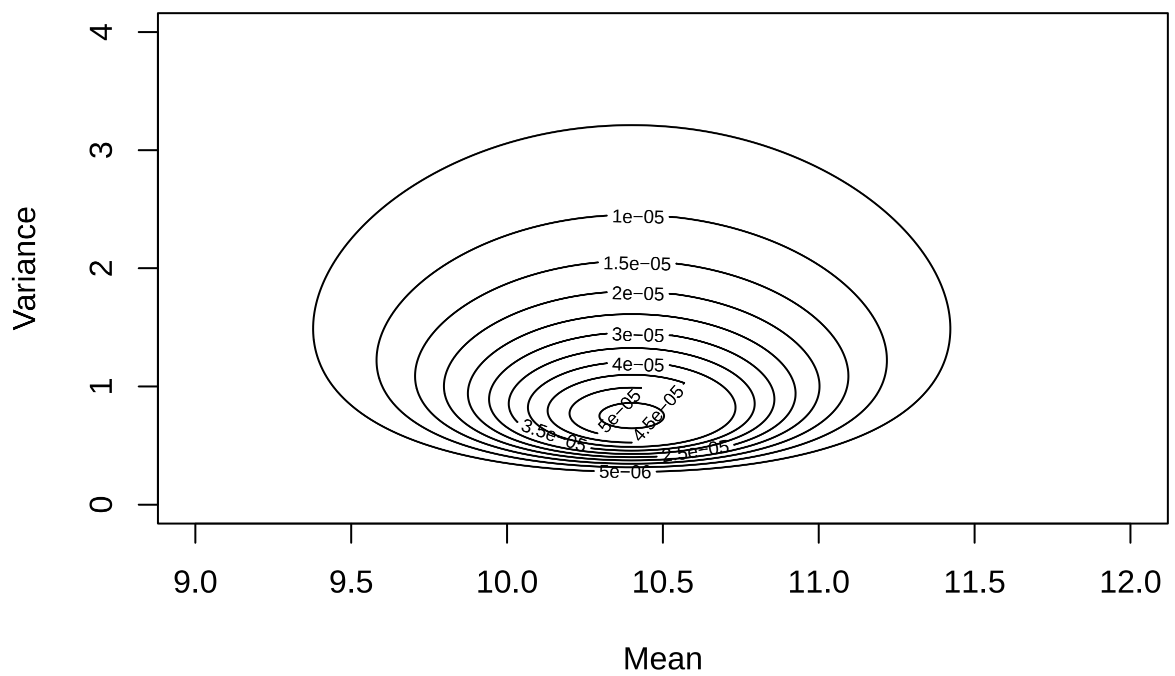
Not Consider Rounding effect _ Method 2 - Grid Approximation

```
pos_prob <- function(mu_1, sig_1){

  out <- sig_1^(-3.5) * exp(-sum((y-mu_1)^2)/(2*sig_1))
  return(out)
}
```

```r
sig_grid <- seq(from = 0.01, to = 5, by = 0.01)
mu_grid <- seq(from = 5, to = 15, by = 0.01)

pos <- matrix(NA, nrow = length(sig_grid), ncol = length(mu_grid))
i <- 1
j <- 1
for (i in 1:length(sig_grid)) {
  for (j in 1:length(mu_grid)) {
    pos[i,j] <- pos_prob(mu_1 = mu_grid[j], sig_1 = sig_grid[i])

  }

}
pos[1,] <- 0
```

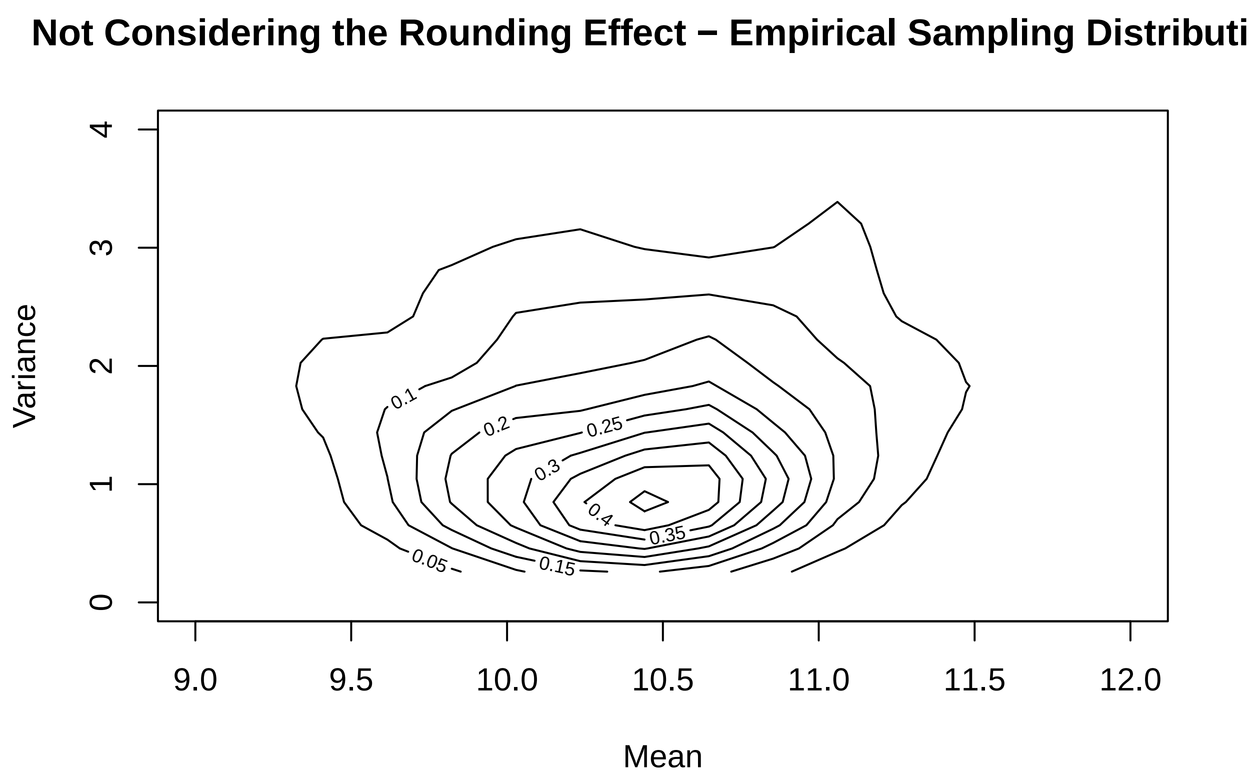
```r
contour(x = mu_grid, y = sig_grid, z = t(pos)/sum(pos, na.rm = TRUE), xlim = c(9, 12), ylim = c(0, 4),
```

Not Considering the Rounding Effect − Grid Approximation




```r
pos_mu_index <- sample(x = 1:length(mu_grid),size = 1000, prob = colSums(pos, na.rm = TRUE), replace =
pos_sig_index <- rep(NA, length(pos_mu_index))
i <- 1
for (i in 1: length(pos_mu_index)) {
  pos_sig_index[i] <- sample(x = 1:length(sig_grid), size = 1, prob = pos[,pos_mu_index[i]], replace =
}
pos_mu <- 5 + 0.01*(pos_mu_index-1)
pos_sig <- 0.01 + 0.01*(pos_sig_index-1)
```

```
den <- MASS::kde2d(x = pos_mu, y = pos_sig)
contour(x = den$x, y = den$y, z = den$z, xlim = c(9, 12), ylim = c(0, 4),
        xlab = "Mean", ylab = "Variance", main = "Not Considering the Rounding Effect - Empirical Sampl
```




```
sum_2 <- matrix( round(
  c(quantile(pos_mu, 0.025), mean(pos_mu) ,quantile(pos_mu, 0.975), var(pos_mu),
    quantile(pos_sig, 0.025), mean(pos_sig) ,quantile(pos_sig, 0.975), var(pos_sig)),2 ) , nrow = 2,
  ncol = 4, byrow = TRUE
)
colnames(sum_2) <- c("2.5% Lower", "Mean", "97.5% Upper", "Variance")
rownames(sum_2) <- c("mu", "sigma")
knitr::kable(sum_2, align = 'c', caption = "Not Considering Rounding Effect - Grid Approximation")
```

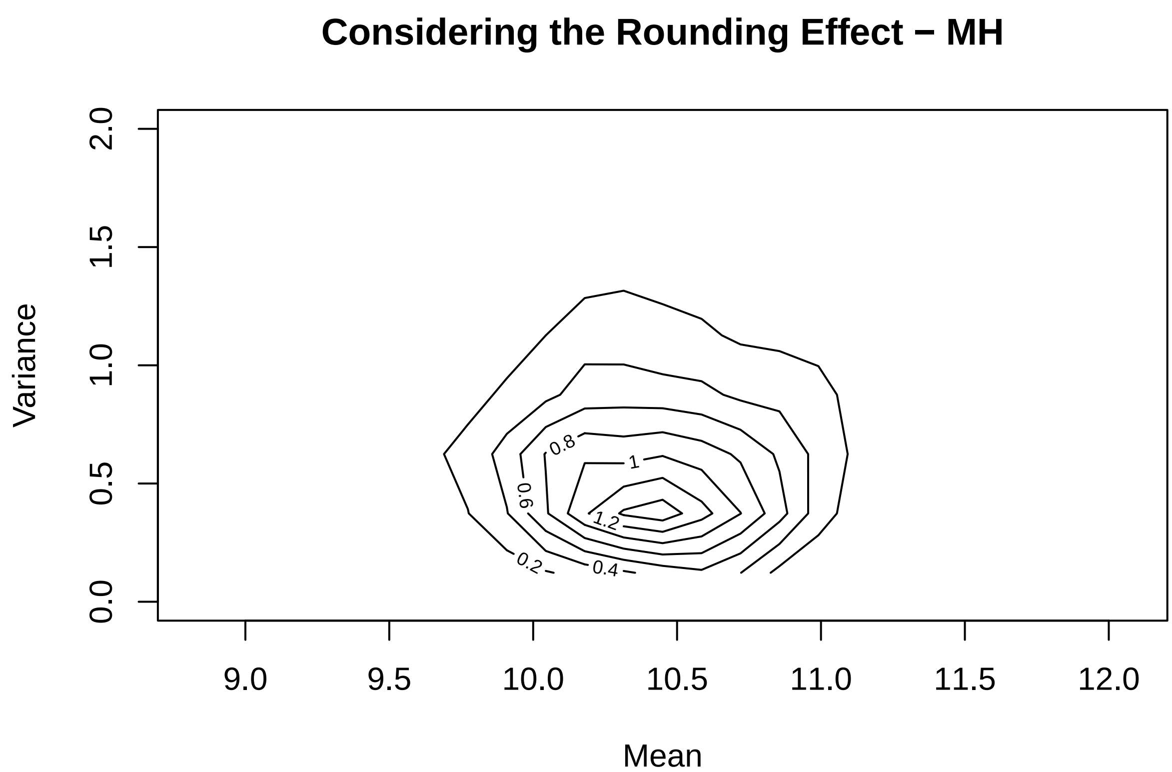
Table 2: Not Considering Rounding Effect - Grid Approximation

	2.5% Lower	Mean	97.5% Upper	Variance
mu	9.34	10.43	11.69	0.34
sigma	0.45	1.69	4.32	1.06

Consider Rounding effect _ Method 1 - MH

```r
set.seed(0)

prob <- function(y, mu, sig){
  out<- rep(NA, length(y))
  j <- 1
  for (j in 1:length(y)) {
    out[j] <- pnorm(y[j]+0.5, mean = mu, sd = sqrt(sig)) - pnorm(y[j]-0.5, mean = mu, sd = sqrt(sig))
  }
  return(out)
}

param <- matrix(NA, nrow = 2, ncol = 10000)

param[,1] <- c(10, 2)
i <- 1
for (i in 1:9999) {
  new_mu <- rnorm(1, mean = param[1,i], sd = 0.3)
  new_sig <- 1/rgamma(1, shape = 1.6, rate = 1.5)
  qold <- dgamma(1/param[2,i], shape = 1.6, rate = 1.5)
  qnew <- dgamma(1/new_sig, shape = 1.6, rate = 1.5)
  pold <- 1/param[2,i] * prod(prob(y = y, mu = param[1,i], sig = param[2,i]))
  pnew <- 1/new_sig * prod(prob(y = y, mu = new_mu, sig = new_sig))
  ptrans <- (pnew * qold) / (pold * qnew)
  index <- runif(1)
  if(index <= ptrans){
    param[1,i+1] <- new_mu
    param[2,i+1] <- new_sig
  }else{
    param[,i+1] <- param[,i]
  }
}
```

```r
#thin <- seq(from = 2000, to = 10000, by = 8)
final <- param[,thin]
den <- MASS::kde2d(x = final[1,], y = final[2,])
contour(x = den$x, y = den$y, z = den$z, main = "Considering the Rounding Effect - MH", xlab = "Mean", y
```




```
mu <- final[1,]
sig <- final[2,]
```

```
sum_3 <- matrix( round(
  c(quantile(mu, 0.025), mean(mu) ,quantile(mu, 0.975), var(mu),
    quantile(sig, 0.025), mean(sig) ,quantile(sig, 0.975), var(sig)),2 ) , nrow = 2,
  ncol = 4, byrow = TRUE
)
colnames(sum_3) <- c("2.5% Lower", "Mean", "97.5% Upper", "Variance")
rownames(sum_3) <- c("mu", "sigma")
knitr::kable(sum_3, align = 'c', caption = "Considering Rounding Effect - MH")
```

Table 3: Considering Rounding Effect - MH

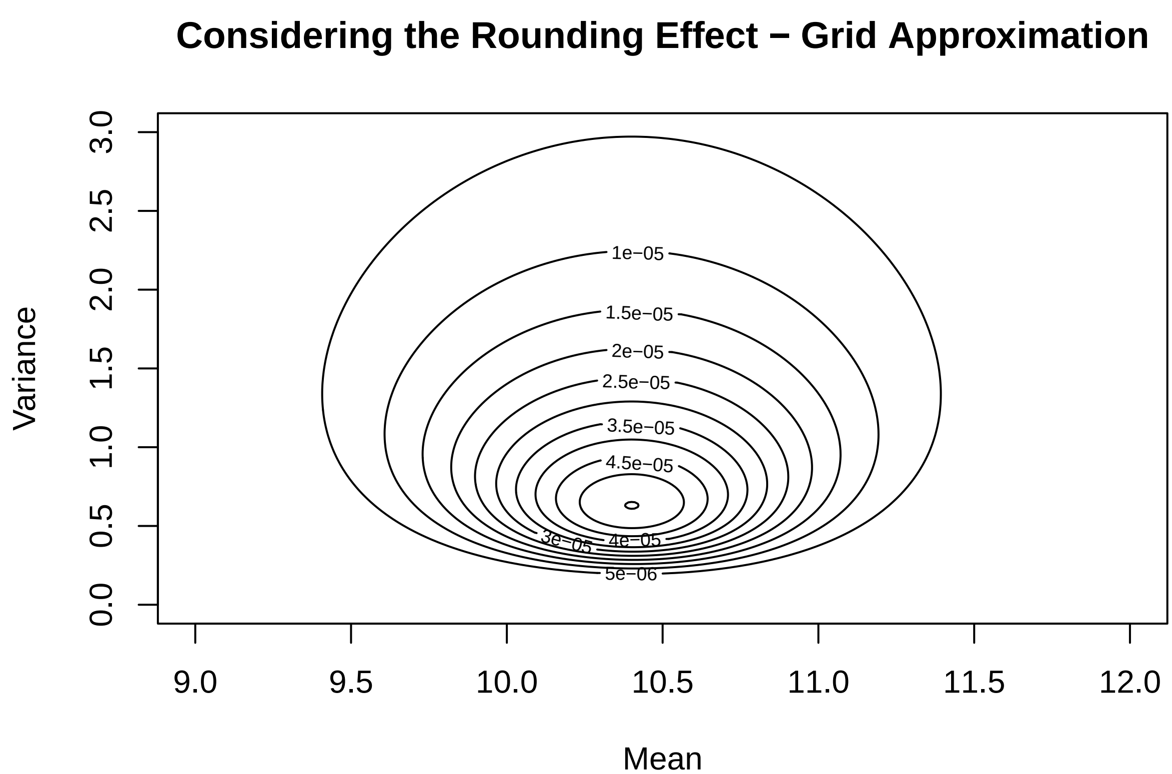
	2.5% Lower	Mean	97.5% Upper	Variance
mu	9.59	10.40	11.21	0.17
sigma	0.18	0.73	2.13	0.31

Consider Rounding effect _ Method 2 - Grid Approximation

```
pos_prob <- function(mu_1, sig_1){

  out <- sig_1^(-1) * prod(prob(y, mu = mu_1, sig = sig_1))
```

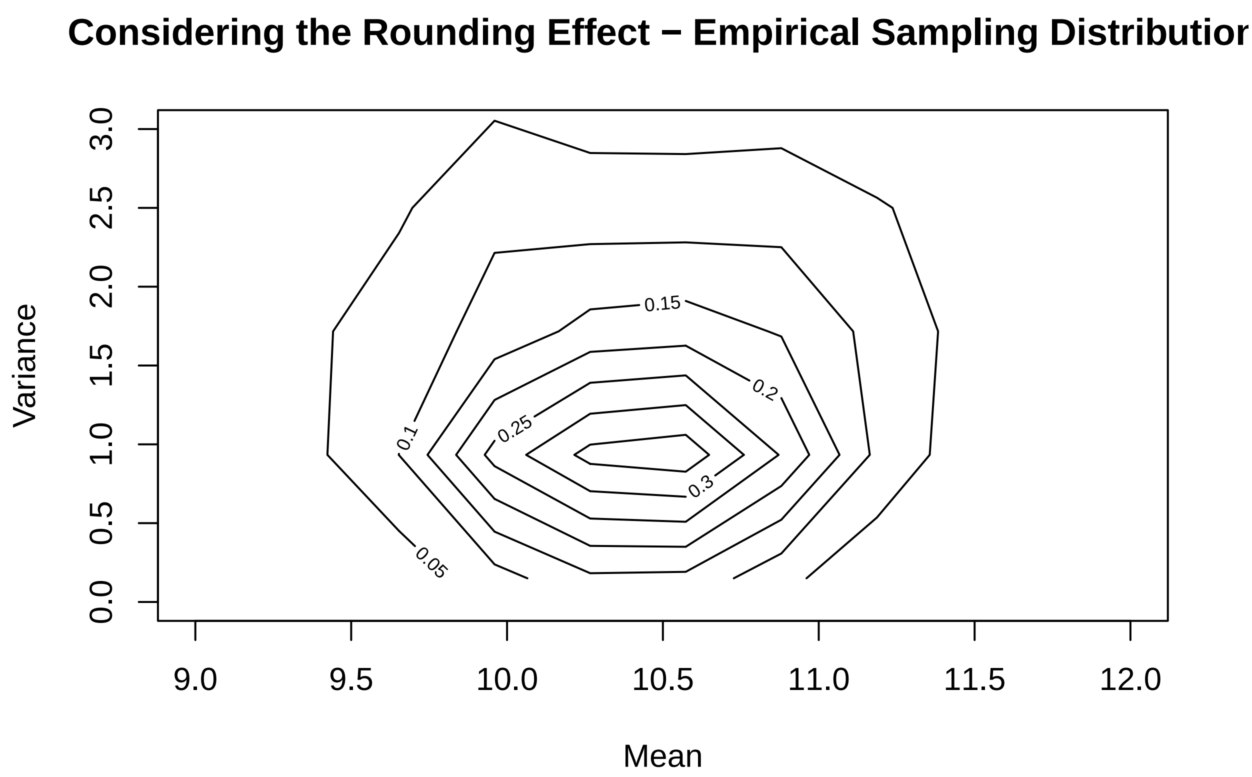
```
  return(out)
}

sig_grid <- seq(from = 0.01, to = 20, by = 0.01)
mu_grid <- seq(from = 5, to = 15, by = 0.01)

pos <- matrix(NA, nrow = length(sig_grid), ncol = length(mu_grid))
i <- 1
j <- 1
for (i in 1:length(sig_grid)) {
  for (j in 1:length(mu_grid)) {
    pos[i,j] <- pos_prob(mu_1 = mu_grid[j], sig_1 = sig_grid[i])

  }

}
pos[1,] <- 0
```

```
contour(x = mu_grid, y = sig_grid, z = t(pos)/sum(pos, na.rm = TRUE), ylim = c(0,3), xlim = c(9,12),  ma
```




```
pos_mu_index <- sample(x = 1:length(mu_grid),size = 1000, prob = colSums(pos, na.rm = TRUE), replace =
pos_sig_index <- rep(NA, length(pos_mu_index))
i <- 1
for (i in 1: length(pos_mu_index)) {
  pos_sig_index[i] <- sample(x = 1:length(sig_grid), size = 1, prob = pos[,pos_mu_index[i]], replace =
```

```
}
pos_mu <- 5 + 0.01*(pos_mu_index-1)
pos_sig <- 0.01 + 0.01*(pos_sig_index-1)

den <- MASS::kde2d(x = pos_mu, y = pos_sig)
contour(x = den$x, y = den$y, z = den$z, main = "Considering the Rounding Effect - Empirical Sampling Di
```




```
sum_4 <- matrix( round(
  c(quantile(pos_mu, 0.025), mean(pos_mu) ,quantile(pos_mu, 0.975), var(pos_mu),
    quantile(pos_sig, 0.025), mean(pos_sig) ,quantile(pos_sig, 0.975), var(pos_sig)),2 ) , nrow = 2,
  ncol = 4, byrow = TRUE
)
colnames(sum_4) <- c("2.5% Lower", "Mean", "97.5% Upper", "Variance")
rownames(sum_4) <- c("mu", "sigma")
knitr::kable(sum_4, align = 'c', caption = "Considering Rounding Effect - Grid Approximation")
```

Table 4: Considering Rounding Effect - Grid Approximation

	2.5% Lower	Mean	97.5% Upper	Variance
mu	9.22	10.42	11.71	0.43
sigma	0.38	2.14	9.19	5.43

Question 7.d: Truncated Normal Distribution

```
library(truncnorm)
```

```
## Warning:   'truncnorm' R 4.1.3
```

```
sample_trunc <- function(n , y){
  out <- matrix(NA, nrow = length(y), ncol = n)
  i <- 1
  j <- 1
  for (i in 1:n) {
    for (j in 1:length(y)) {
      out[j,i] <- rtruncnorm(1, a = y[j]-0.5, b = y[j] + 0.5, mean = 10.4, sd = sqrt(2.14))

    }

  }
  return(out)
}
pos_sample <- sample_trunc(10000, y = y)
mean((pos_sample[1,] - pos_sample[2,])^2)
```

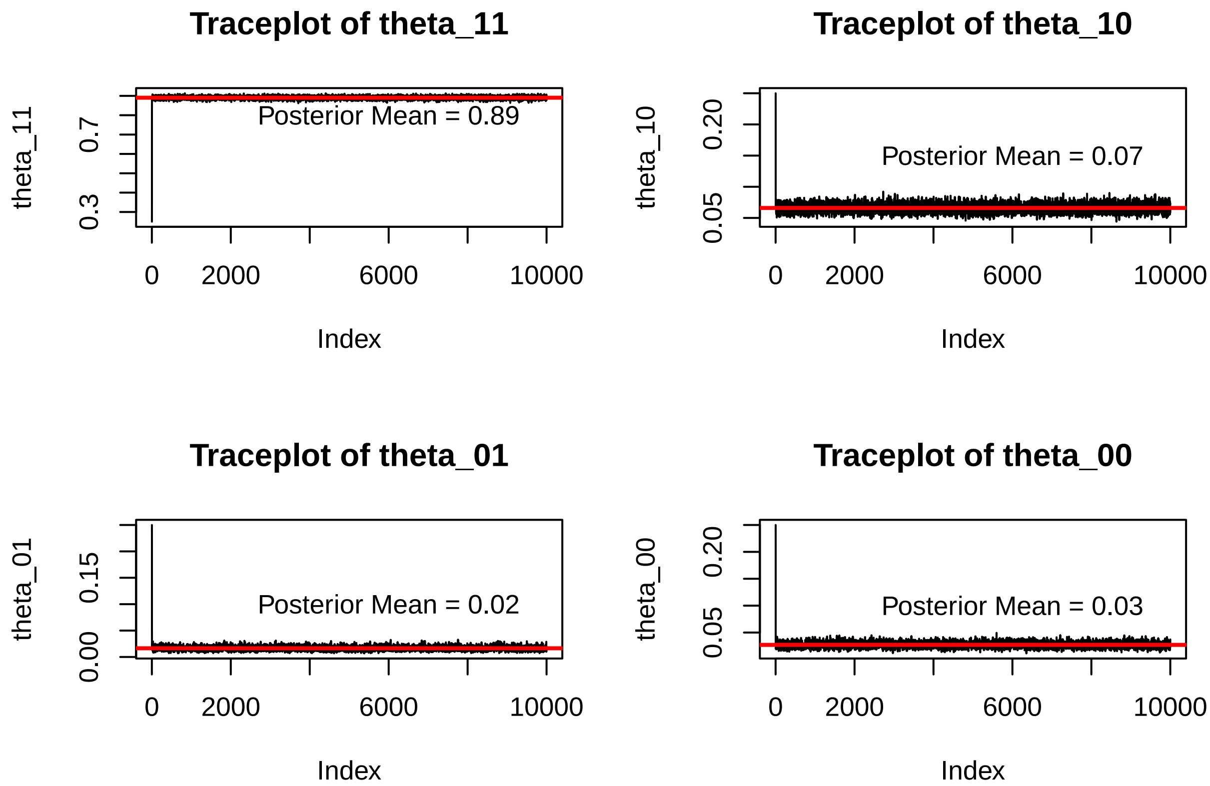
```
## [1] 0.1671331
```

Question 9.c: Dirichlet Distribution and Multinomial Distribution

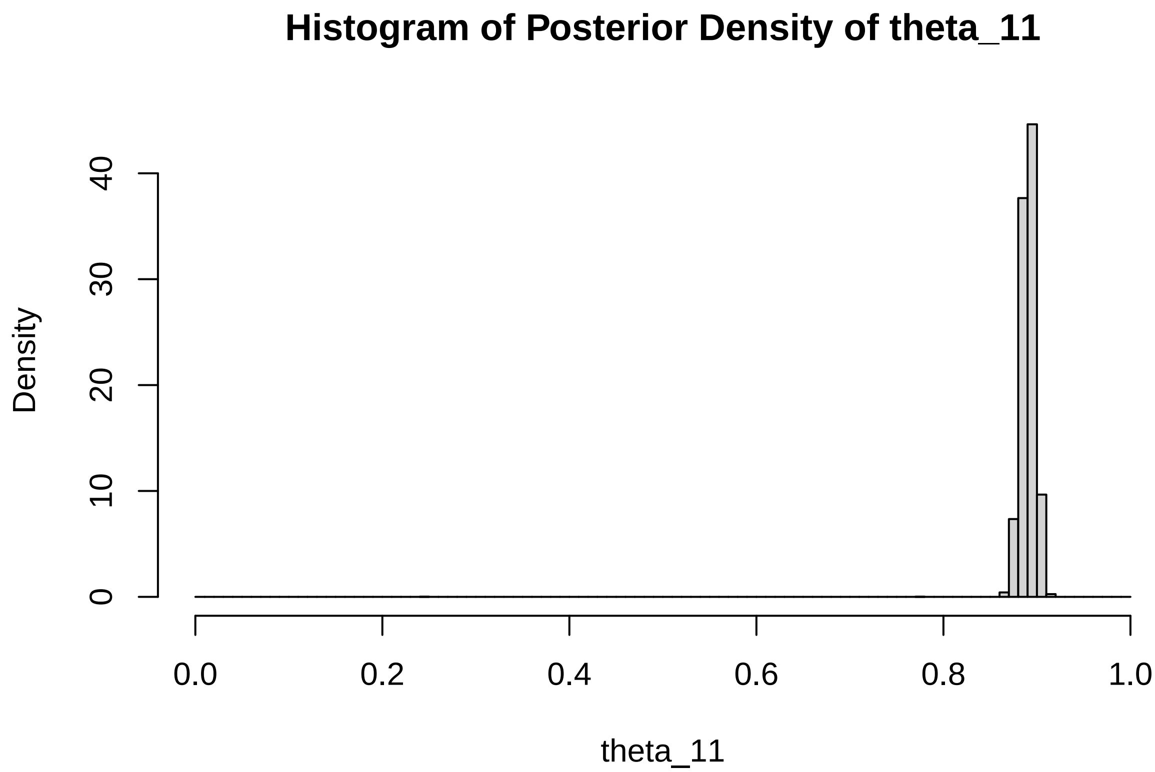
```
x_11 <- 1439
x_10 <- 78
x_01 <- 16
x_00 <- 16
x_1d <- 159
x_0d <- 32
x_d1 <- 144
x_d0 <- 54
x_dd <- 136
theta <- matrix(NA, nrow = 4, ncol = 10000)
theta[,1] <- rep(0.25, 4)
i <- 1
set.seed(0)
for (i in 1:9999) {
  x_1d_11 <- rbinom(n = 1, size = x_1d, prob = theta[1,i]/(theta[1,i]+theta[2,i]))
  x_1d_10 <- x_1d - x_1d_11
  x_d1_11 <- rbinom(n = 1, size = x_d1, prob = theta[1,i]/(theta[1,i]+theta[3,i]))
  x_d1_01 <- x_d1 - x_d1_11
  x_0d_01 <- rbinom(n = 1, size = x_0d, prob = theta[3,i]/(theta[3,i]+theta[4,i]))
  x_0d_00 <- x_0d - x_0d_01
  x_d0_10 <- rbinom(n = 1, size = x_d0, prob = theta[2,i]/(theta[2,i]+theta[4,i]))
  x_d0_00 <- x_d0 - x_d0_10

  x_dd_all <- rmultinom(n = 1, size = x_dd, prob = c(theta[1,i]/sum(theta[,i]),theta[2,i]/sum(theta[,i]
                                                     theta[3,i]/sum(theta[,i]),theta[4,i]/sum(theta[,i]
  x_dd_11 <- x_dd_all[1]
  x_dd_10 <- x_dd_all[2]
  x_dd_01 <- x_dd_all[3]
```

```r
  x_dd_00 <- x_dd_all[4]

  y_11 <- x_11 + x_1d_11 + x_d1_11 + x_dd_11
  y_10 <- x_10 + x_1d_10 + x_d0_10 + x_dd_10
  y_01 <- x_01 + x_d1_01 + x_0d_01 + x_dd_01
  y_00 <- x_00 + x_0d_00 + x_d0_00 + x_dd_00
  theta[,i+1] <- gtools::rdirichlet(1, alpha = c(y_11+1, y_10+1, y_01+1, y_00+1))
}
```

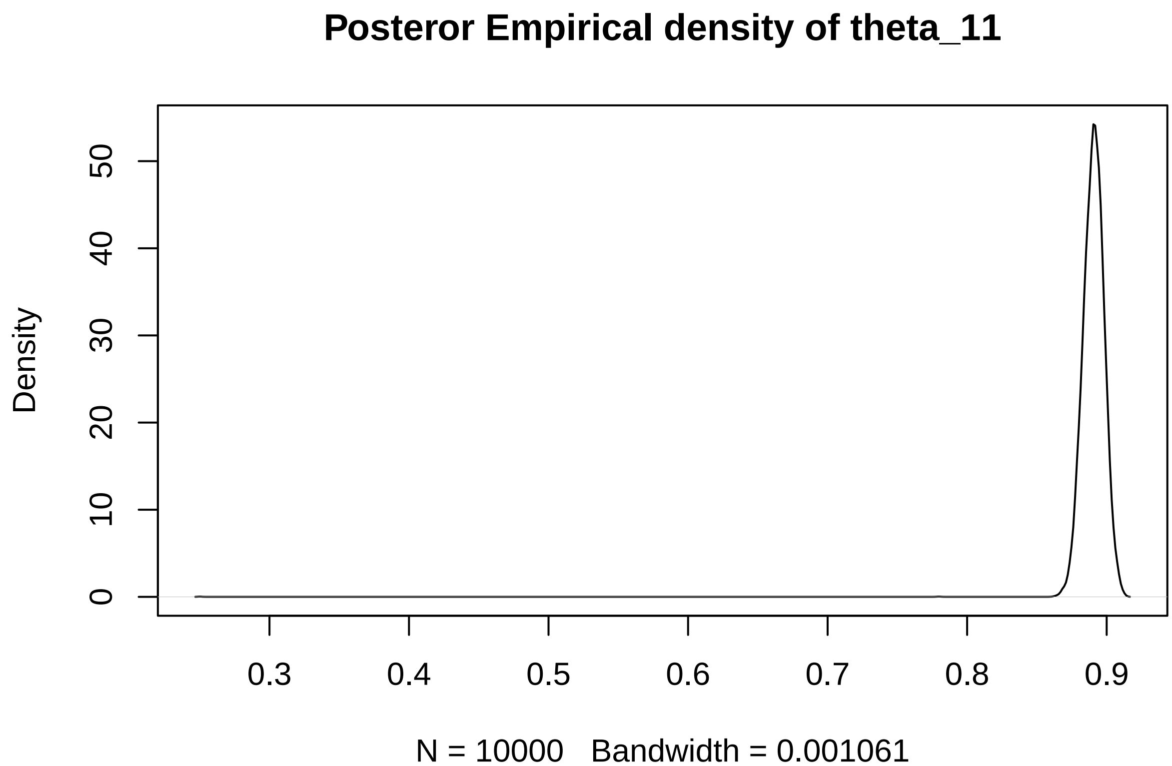
```r
theta_11 <- theta[1,]
theta_10 <- theta[2,]
theta_01 <- theta[3,]
theta_00 <- theta[4,]
par(mfrow = c(2,2))
plot(theta_11, type = 'l', ylab = expression(theta_11), main = "Traceplot of theta_11"  )
abline(h = mean(theta_11), col = "red", lwd = 2)
text(x = 6000, y = 0.8, paste("Posterior Mean =",round(mean(theta_11),2)))

plot(theta_10, type = 'l', ylab = expression(theta_10), main = "Traceplot of theta_10"  )
abline(h = mean(theta_10), col = "red", lwd = 2)
text(x = 6000, y = 0.15, paste("Posterior Mean =",round(mean(theta_10),2)))

plot(theta_01, type = 'l', ylab = expression(theta_01), main = "Traceplot of theta_01"  )
abline(h = mean(theta_01), col = "red", lwd = 2)
text(x = 6000, y = 0.1, paste("Posterior Mean =",round(mean(theta_01),2)))

plot(theta_00, type = 'l', ylab = expression(theta_00), main = "Traceplot of theta_00"  )
abline(h = mean(theta_00), col = "red", lwd = 2)
text(x = 6000, y = 0.1, paste("Posterior Mean =",round(mean(theta_00),2)))
```




```
hist(theta_11, probability = TRUE, breaks = seq(from = 0, to = 1, by = 0.01), main = "Histogram of Poste
```




```
plot(density(theta_11), type = 'l', main = "Posteror Empirical density of theta_11")
```


Posteror Empirical density of theta_11
Density
0
10
20
30
40
50
0.3
0.4
0.5
0.6
0.7
0.8
0.9
N = 10000   Bandwidth = 0.001061