# MCMC_HW1

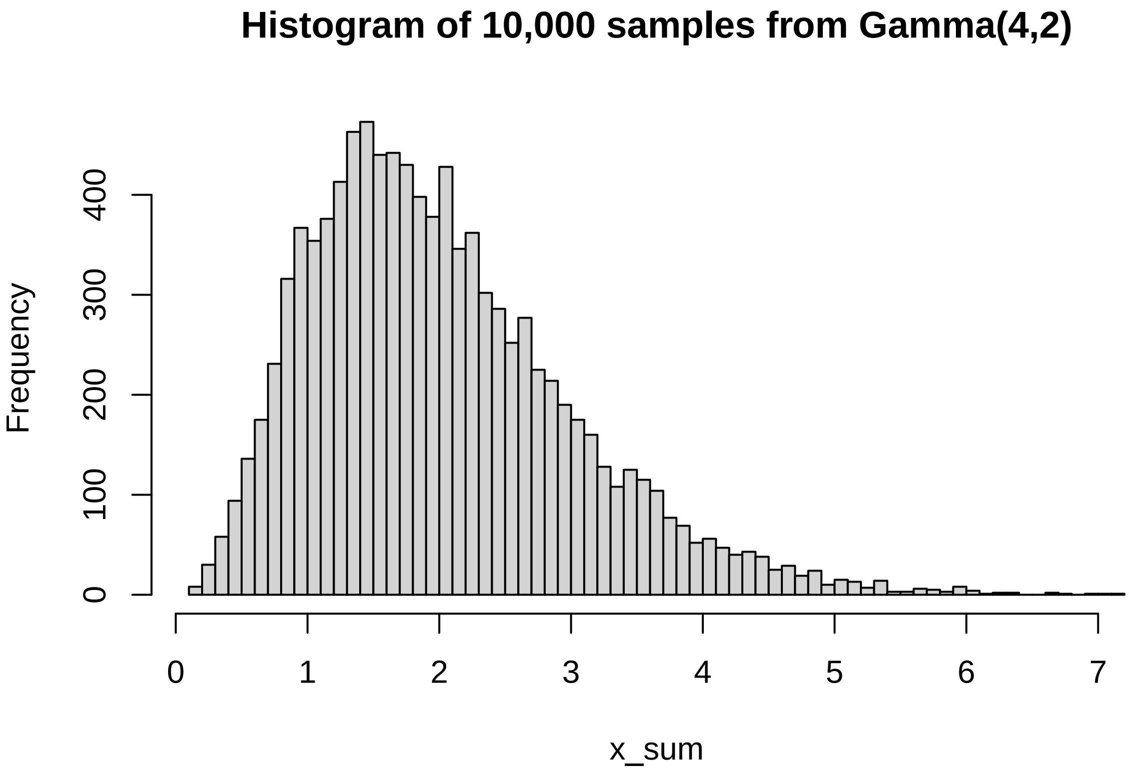
Suman

03/10/2021

### [HW1] Generate 10,000 samples from Gamma(4,2) and Gamma(6,3) and then draw histograms.

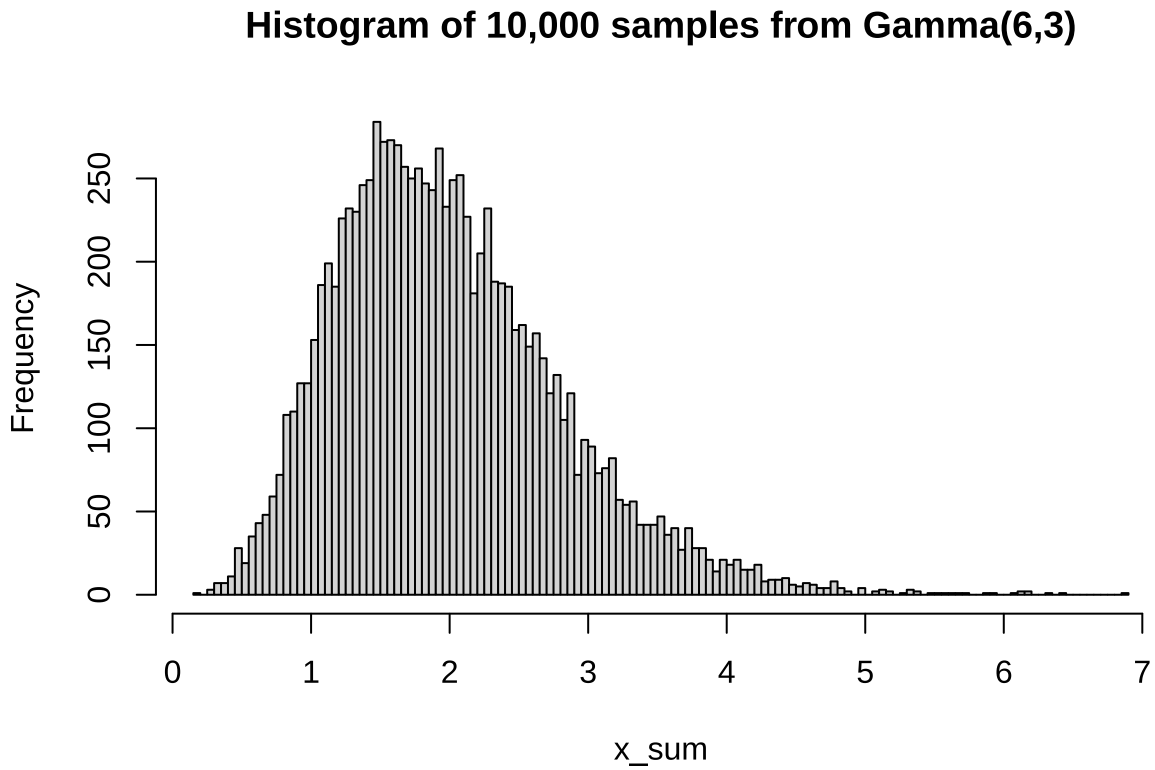
$$\begin{aligned} Gamma(4,2) &= \sum_{i=1}^{4} Gamma(1,2) \quad (i.i.d.) \\ &= \sum_{i=1}^{4} Exp(2) \end{aligned}$$

```r
alpha = 4 ; beta = 2
u = matrix(runif(10000*alpha), ncol=alpha)

# Using inverse cdf method for exponential distribution with parmater 2
x = -(1/beta)*log(1-u)

# Sum of 4 exp(2)
x_sum = rowSums(x)

# Histogram
hist(x_sum, nclass=100, main="Histogram of 10,000 samples from Gamma(4,2)")
```



$$
\begin{aligned}
Gamma(6,3) &= \sum_{i=1}^{6} Gamma(1,3) \quad (i.i.d.) \\
&= \sum_{i=1}^{6} Exp(3)
\end{aligned}
$$

```r
alpha=6 ; beta=3
u = matrix(runif(10000*alpha), ncol=alpha)

# Using inverse cdf method for exp(3)
x = -(1/beta)*log(1-u)

# Sum of 6 exp(3)
x_sum = rowSums(x)

# Histogram
hist(x_sum, nclass=100, main="Histogram of 10,000 samples from Gamma(6,3)")
```



First, we know that the pdf of Cauchy(0,1) is

$$f(x) = \frac{1}{\pi(1+x^2)} \quad x \in \Re$$

Then the cdf(cumulative distribution function) of the Cauchy(0,1) is

$$\begin{aligned}\int_{-\infty}^{x} \frac{1}{\pi(1+t^2)} dt &= \frac{1}{\pi} arctan(t) \mid_{-\infty}^{x} \\ &= \frac{1}{\pi} arctan(x) + \frac{1}{2}\end{aligned}$$

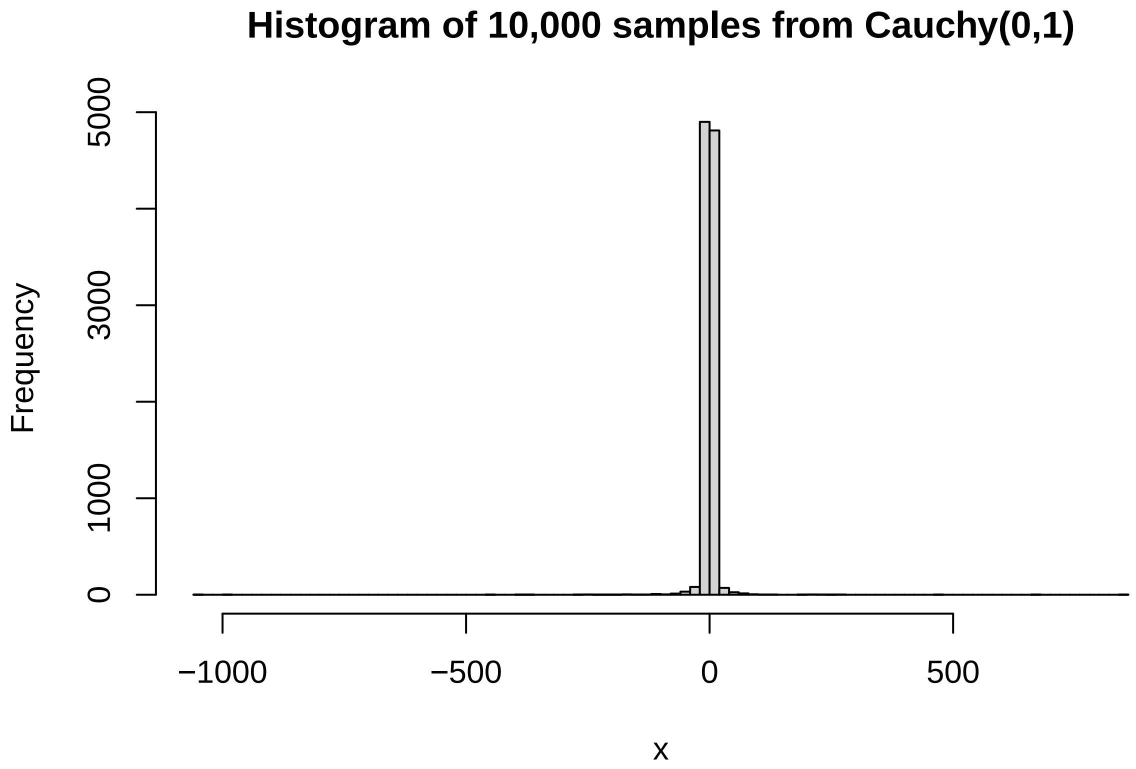
By using inverse cdf method, we can get samples from Cauchy(0,1)

$u = \frac{1}{\pi} arctan(x) + \frac{1}{2}$

$x = tan(\pi(u - \frac{1}{2}))$

In this example, $\pi$ is regarded as 3.14.

```
u = runif(10000)
pi = 3.14
x = tan(pi*(u-1/2))
hist(x, nclass=100, main="Histogram of 10,000 samples from Cauchy(0,1)")
```



### [HW 2] Suppose our target density is the triangle density where

$$F(x) = \begin{cases} 0 & if\ x < 0 \\ 4x^2 & if\ 0 \le x < 0.25 \\ \frac{8}{3}x - \frac{4}{3}x^2 - \frac{1}{3} & if\ 0.25 \le x \le 1 \\ 1 & if\ x > 1 \end{cases}$$

#### (a) Generate random number using inverse-CDF

For using inverse-CDF, we need to transform F(x) to be inversed.\ When we set $F(x) = u$ and solve for $x$,

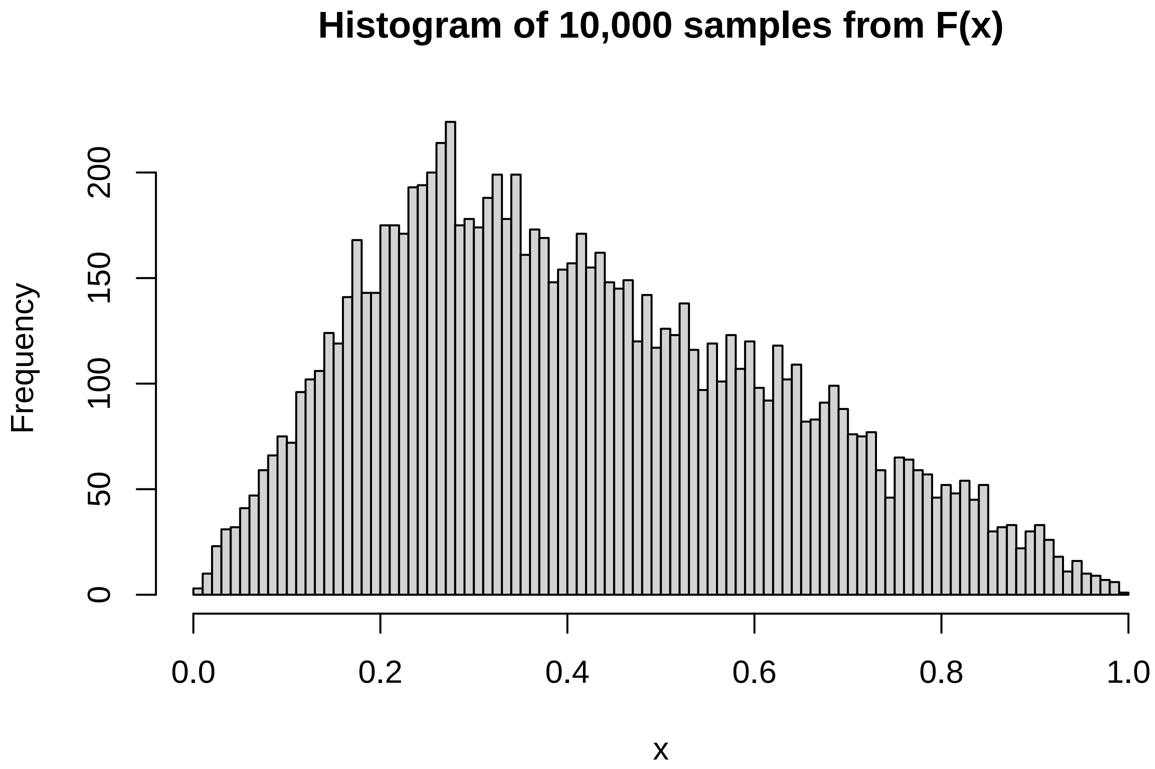
$$F^{-1}(u) = \begin{cases} \frac{\sqrt{u}}{2} & if\ 0 \le u < 0.25 \\ 1 - \frac{\sqrt{3(1-u)}}{2} & if\ 0.25 \le u \le 1 \end{cases}$$

Even if we can not get the solutions when u is not in $[0, 1]$, it is okay that u follows $Unif(0, 1)$

```
# sample size
n <- 10000

# inverse CDF
u <- runif(10000)
inverse.cdf <- function(u) {
    if (u >= 0 && u < 0.25)
        sqrt(u)/2
    else if (u >=0.25 && u <=1)
        1 - sqrt(3*(1-u))/2 }

# Generate random number
x <- unlist(lapply(u, inverse.cdf))

# Show the 100 samples of generated random numbers
head(x, 100)
```

```
##   [1] 0.20814105 0.61108643 0.29211531 0.14630157 0.23486112 0.42364065
##   [7] 0.24733409 0.45849270 0.08308455 0.16850609 0.39011336 0.49863209
##  [13] 0.17032093 0.56792003 0.91634964 0.17243318 0.76699854 0.50966552
##  [19] 0.52519733 0.27620533 0.72165099 0.23534501 0.23613544 0.06227682
##  [25] 0.10941467 0.37278250 0.19221072 0.77164032 0.17067520 0.16246924
##  [31] 0.66922456 0.50049809 0.35202266 0.30101543 0.80772751 0.38443481
##  [37] 0.25279995 0.63200877 0.36885807 0.25060121 0.21869343 0.65996184
##  [43] 0.43985288 0.51749649 0.49555230 0.24948503 0.16071857 0.44970503
##  [49] 0.52673392 0.53294773 0.37328336 0.38753051 0.26141485 0.42810274
##  [55] 0.61048343 0.13179562 0.29532438 0.39071377 0.25140601 0.51517942
##  [61] 0.60640313 0.58836616 0.20355889 0.98047309 0.61929964 0.29273074
##  [67] 0.35476285 0.22064090 0.38963882 0.09915655 0.29234311 0.38599288
##  [73] 0.33271217 0.27380667 0.40000298 0.36695679 0.59964280 0.23956264
##  [79] 0.55364429 0.48453358 0.69528994 0.88783060 0.79643044 0.25358792
##  [85] 0.17167107 0.63433101 0.23793477 0.14885550 0.24499302 0.13326159
##  [91] 0.67326870 0.42640051 0.43811386 0.58092743 0.33490881 0.54214656
##  [97] 0.41692403 0.43734901 0.43264543 0.67689936
```

```
# Histogram
hist(x, nclass=100, main="Histogram of 10,000 samples from F(x)")
```



**(b) Suppose g follows unif(0,1) and e = 3g. Generate samples using rejection sampling.**

First, we should get the target density $f(x)$ from $F(x)$.

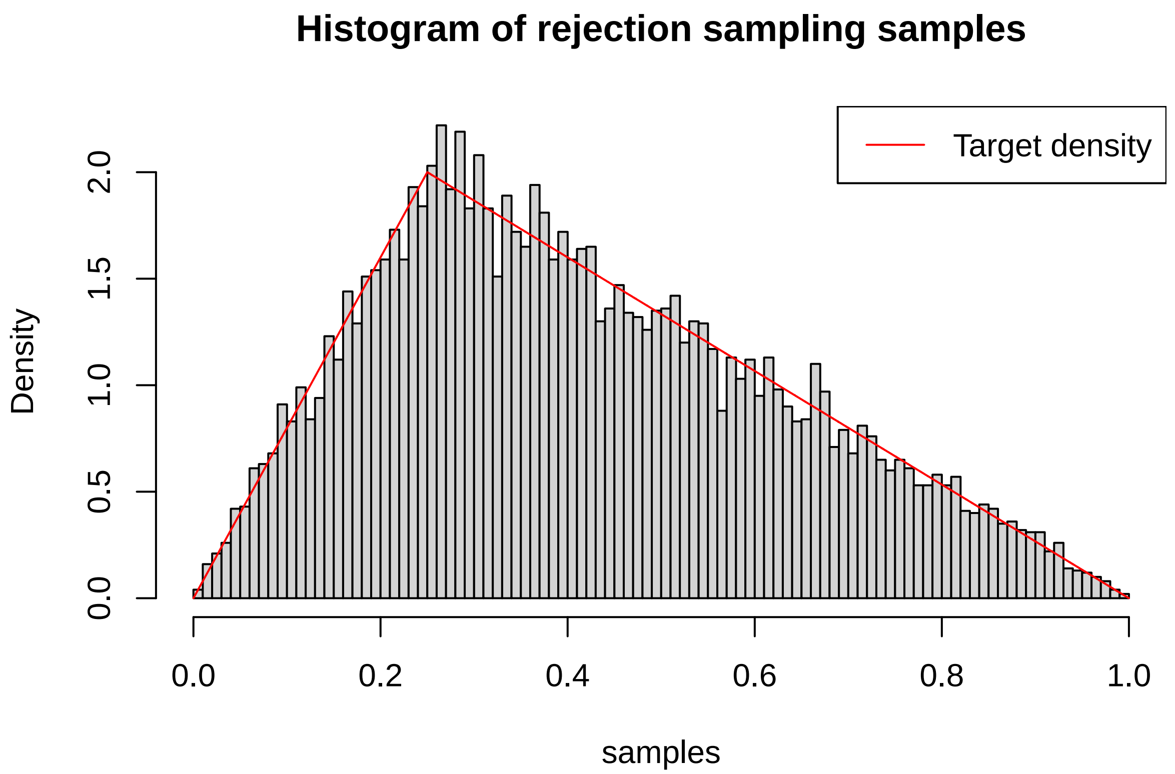
$$f(x) = \begin{cases} 8x & if\ 0 \leq x < 0.25 \\ \frac{8}{3} - \frac{8}{3}x & if\ 0.25 \leq x \leq 1 \\ 0 & o.w. \end{cases}$$

Using rejection sampling method…

```
## 1. Setup

# 1-0. desired sample size and sample space
desired_size <- 10000
samples <- c()

# 1-1. target density f(x)

target.x <- function(x) {
    if ((x >=0) && (x < 0.25))
        8*x
    else if ((x >=0.25) && (x<=1))
        8/3 - 8/3*x
    else
        0
```

```r
}

# 1-2. proposal density
proposal.x <- function(x) {
    if (x>=0 && x<=1)
        1
    else
        0
}

# 1-3. envelope density e(x) = 3g(x)
envelope.x <- function(x) {
    if (x>=0 && x<=1)
        3
    else
        0
}


## 2. Rejection sampling

# 2-1. Sample y from g(x)
# y <- runif(1)

# 2-2. Sample u from Unif(0,1)
#u <- runif(1)

# 2-3. Calculation of rejection probability
# rej_prob <- target.x / envelope.x

### reject y if u > rej_prob

# This process should be repeated until we have samples of the desired size(10,000)
curr_size <- 0
while(curr_size < desired_size) {
    y <- runif(1)
    u <- runif(1)
    rej_prob <- target.x(y) / envelope.x(y)

    if (u <= rej_prob) {
        samples <- c(samples, y)
        curr_size <- curr_size + 1
    }
}

# 3. Compare the target density and distribution of samples of rejection sampling.
value <- c()
res <- c()

for (i in sort(runif(10000)))
{res <- c(res, target.x(i))
value <- c(value, i)}
```

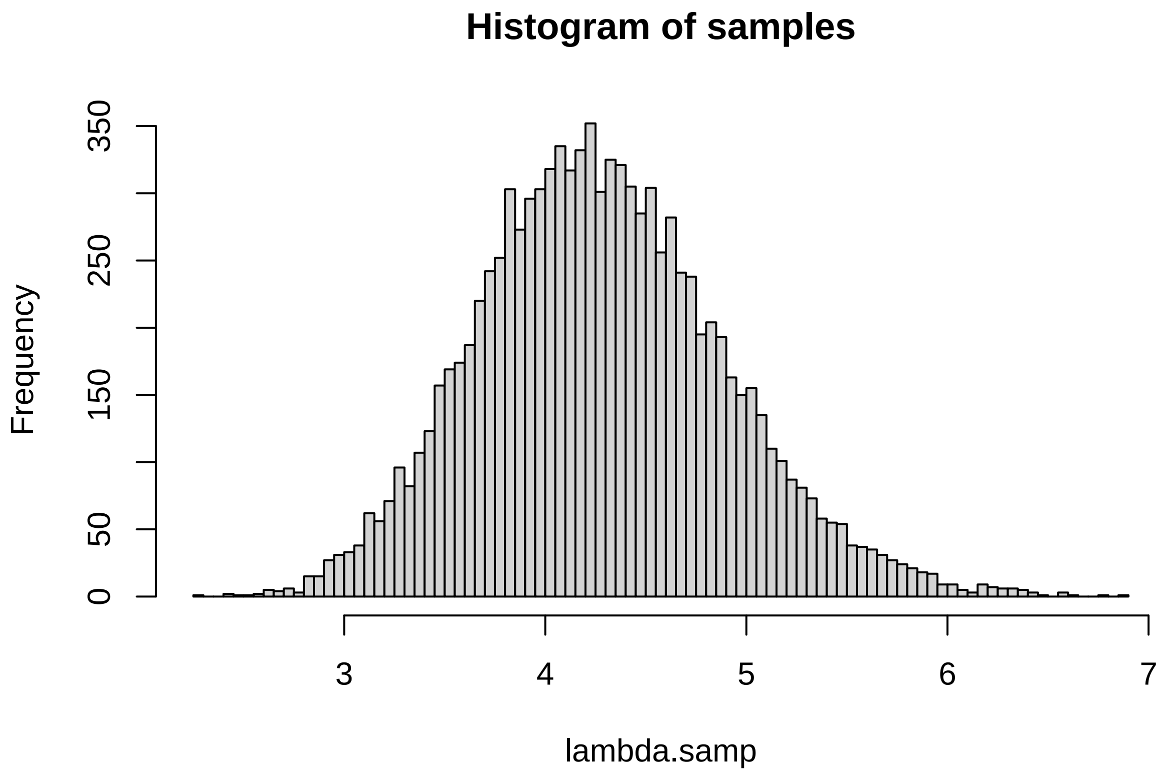
```
hist(samples, nclass= 100, freq=FALSE, main="Histogram of rejection sampling samples")
lines(value ,res, type="l", col="red")
legend("topright",c("Target density"), col=("red"), lty=1)
```



We can get samples which have almost same distribution with target density by using rejection sampling method. Also, the result of rejection sampling is almost same with the result of invese-CDF method.

### [Example] Sampling a Bayesian Posterior

```
# set values
x <- c(8,3,4,3,1,7,2,6,2,7)
n <- 10000
lambda.samp <- rep(NA, n)
xbar = mean(x)

# proposal density
iter = 1
total = 1

while(iter <= n){
  lambda = exp(rnorm(1, log(4), 0.5))
  u = runif(1,0,1)
  ratio = exp(sum(dpois(x, lambda, log=TRUE)) - sum(dpois(x, xbar, log=TRUE)))
  if (u <= ratio){
    lambda.samp[iter] = lambda
    iter = iter+1
  }
  total = total+1
}

hist(lambda.samp, nclass=100, main= "Histogram of samples")
```



```
n/total
```

```
## [1] 0.2898635
```