

Atlas-PS 5

David Atlas

9/29/2018

Problem 1

a)

We show that $\text{Var}_g w^*(X) < M - 1$:

$$\begin{aligned}\text{Var}_g w^*(X) &= E[w^*(x)^2] - E[w^*(x)]^2 \\ &= E\left[\frac{f(x)^2}{g(x)^2}\right] - E\left[\frac{f(x)}{g(x)}\right]^2 \\ &= \int_{-\infty}^{\infty} \frac{f(x)^2}{g(x)^2} g(x) dx - \int_{-\infty}^{\infty} \frac{f(x)}{g(x)} g(x) dx \\ &= \int_{-\infty}^{\infty} f(x) \frac{f(x)}{g(x)} dx - \int_{-\infty}^{\infty} f(x) dx.\end{aligned}$$

Next, we note that $\frac{f(x)}{g(x)} < M$, by definition and $\int_{-\infty}^{\infty} f(x) dx = 1$, as $f(x)$ is a probability distribution, which by definition must integrate to 1 over the support. Therefore, we can write:

$$\begin{aligned}\int_{-\infty}^{\infty} f(x) \frac{f(x)}{g(x)} dx - \int_{-\infty}^{\infty} f(x) dx &< \int_{-\infty}^{\infty} M f(x) dx - 1 \\ &= M \int_{-\infty}^{\infty} f(x) dx - 1 \\ &= M - 1.\end{aligned}$$

Therefore, we can say that $\text{Var}_g w^*(x) < M - 1$.

b)

Next, we want to show that $\text{Var}_g(w^*(X)h(X))$ is finite, given that $\text{Var}_g(h(X))$ is finite.

We first reference the Cauchy-Schwarz Inequality, with the inner product being defined as the expected value with respect to g . Therefore, $\langle X, 1 \rangle = E[X]$ implies that $E[X]^2 \leq E[X^2]$. As the variance of both $h(x)$ and $w^*(x)$ is finite with respect to g , the second moment ($E[X^2]$) must be finite as well (this follows from the property of the real space that it is closed under addition/subtraction). Therefore, we know that $E[h(X)]^2$ and $E[w^*(X)]^2$ are both finite.

Next, we rewrite $\text{Var}(w^*(X)h(X))$. (Not going to prove this equality; it's readily available on the Wikipedia page for variance under the heading of "Product of independent variables"). We also assume the $h(X)$ and $w^*(x)$ are independent (otherwise, this equality doesn't appear to be true, as higher order moments must be finite, which we cannot show with the given information).

$$\text{Var}(w^*(X), h(X)) = E[w^*(X)^2]E[h(X)^2] - E[w^*(X)]^2E[h(X)]^2.$$

Given that both $w^*(x)$ and $h(x)$ have finite variance, both of their second order moments must be finite. Therefore, we can say that all 4 of the terms above are finite, and under the basic properties of real numbers, the difference of their products must also be finite. Therefore $\text{Var}(w^*(X), h(X))$ is finite.

Problem 2

We sample the standard normal distribution using a normal distribution with mean 1 and standard deviation 2. We define our functions as follows, where ϕ is the normal density:

$$\begin{aligned} h(x) &= x \\ f(x) &= \phi(x, 0, 1) \\ g(x) &= \phi(x, 1, 2). \end{aligned}$$

Note that $h(x) = x$ because we are simply looking for the expected value of $f(x)$, or $E_f[X]$.

```
# We define the target and importance sampling functions
f <- function(x) dnorm(x, mean=0, sd=1)
g <- function(x) dnorm(x, mean=1, sd=2)
h <- function(x) x

set.seed(73)
n_obs <- 1000
X <- rnorm(n=n_obs, mean=1, sd=2)
X_direct <- rnorm(n=n_obs, mean=0, sd=1)

print(paste0("Weighted Mean: ", round(mean(h(X) * f(X) / g(X)), 4)))

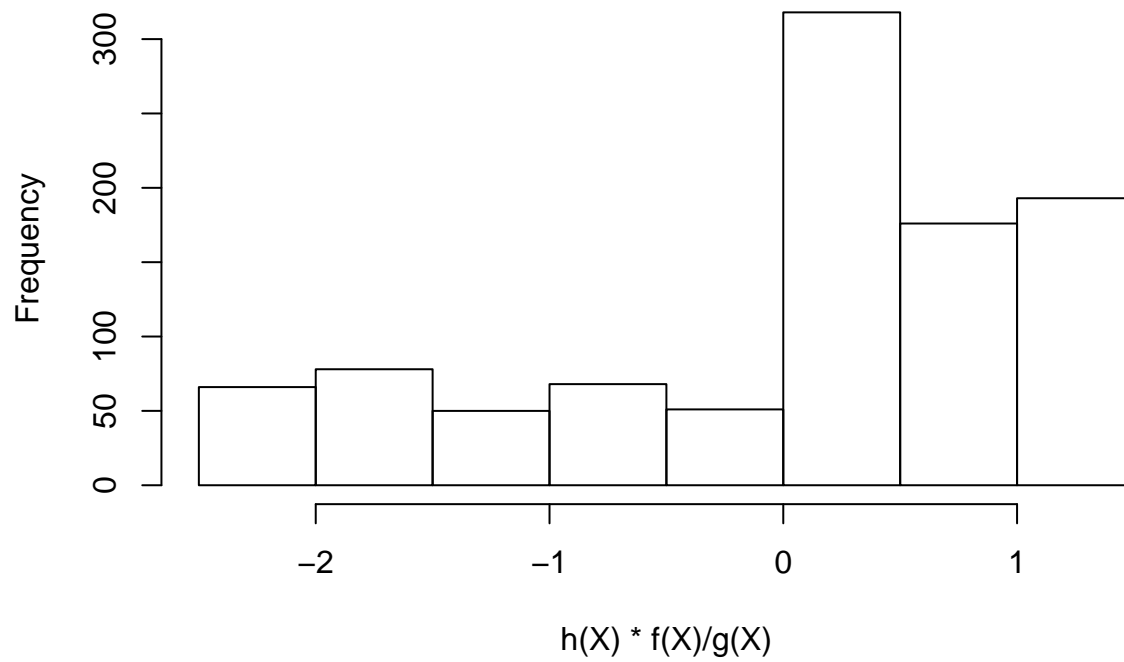
## [1] "Weighted Mean: 0.0012"

print(paste0("Weighted SD: ", round(sd(h(X) * f(X) / g(X)), 4)))

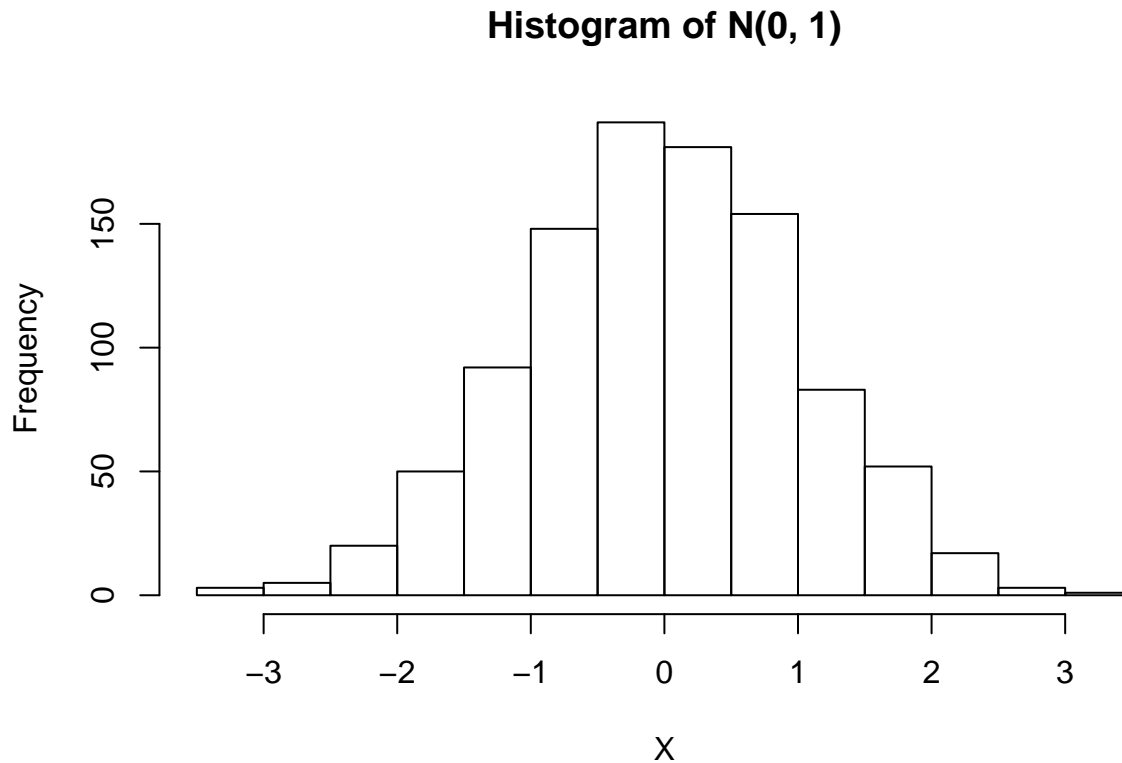
## [1] "Weighted SD: 1.0173"

hist(h(X) * f(X) / g(X), main="Histogram of Importance Sampling Observations")
```

Histogram of Importance Sampling Observations



```
hist(X_direct, main="Histogram of N(0, 1)", xlab="X")
```



We see that the weighted mean and variance found via importance sampling is very close to our target distribution. However, the histogram shows that the observations themselves do not appear to be normally distributed.

Problem 3

a)

We define the transition probability matrix.

$$P = \begin{matrix} & \begin{matrix} \text{Positive Tomorrow} & \text{Negative Tomorrow} & \text{It Could Be Worse Tomorrow} \end{matrix} \\ \begin{matrix} \text{Positive Today} \\ \text{Negative Today} \\ \text{It Could Be Worse Today} \end{matrix} & \begin{bmatrix} 0 & .5 & .5 \\ .25 & .5 & .25 \\ .25 & .25 & .5 \end{bmatrix} \end{matrix}$$

b)

We solve $\pi P = \pi$ by multiplying the matrices to get the following system of equations, where the last one enforces the summing of probabilities to 1.

$$\begin{aligned}
-\pi_p + .5\pi_n + .5\pi_w &= 0 \\
.25\pi_p + -.5\pi_n + .25\pi_w &= 0 \\
.25\pi_p + .25\pi_n + -.5\pi_w &= 0 \\
\pi_p + \pi_n + \pi_w &= 1
\end{aligned}$$

We can solve this system of equations using many techniques, but in the spirit of simulation, we will solve it with the recursive equation $\pi_t = \pi_{t-1}P$:

```
pi0 <- c(1/3, 1/3, 1/3)
pit <- c(1, 1, 1)
P <- matrix(c(0, .5, .5, .25, .5, .25, .25, .25, .5), ncol=3, byrow=TRUE)
while (sum(abs(pi0 - pit)) > .0001){
  pi0 <- pit
  pit <- pi0 %*% P
}
solution <- data.frame(pit %*% P)
colnames(solution) <- c("P", "N", "W")
# Print the normalized solutions
knitr::kable(round(solution / sum(solution), 3))
```

P	N	W
0.2	0.4	0.4

We see that in the long run, 60% of the time, the general population will have a non-negative opinion of the government.

Problem 4

a)

We implement the Metropolis-Hastings Algorithm to generate observations from a mixing distribution

$$f(x) = \delta N(7, .5^2) + (1 - \delta)N(10, .5^2).$$

We use

$$g(x^*|x^{(t)}) = \phi(x^*, x^{(t)}, .01^2)$$

as the proposal density, thereby drawing proposed observations from $N(x^{(t)}, .01^2)$.

We use 0, 7, 15 as the set of starting points x_0 .

```
set.seed(73)

# We implement our density functions f and g
f <- function(x, lambda=.7) lambda * dnorm(x, 7, .5^2) + (1 - lambda) * dnorm(x, 10, .5^2)
g <- function(x, x0) dnorm(x, mean=x0, sd=.01^2)

# We implement our random distribution to draw from (the density is g)
rg <- function(x0) rnorm(n=1, mean=x0, sd=.01^2)
```

```

# Helper function to determine whether or not to keep the random draw
find_xt <- function(xstar, xt, mh_ratio){
  cutoff <- runif(n=1, min=0, max=1) # Draw from U(0, 1)
  return(ifelse(cutoff < min(mh_ratio, 1), xstar, xt))
}

# MH Algorithm
metropolis_hastings <- function(g, rg, f, x0, iterations=10000){
  path <- rep(0, iterations) # Vector to hold path

  # Loop through iterations
  for (i in 1:iterations){
    # Get a random draw from g
    xstar <- rg(x0)

    # Calculate the MH ratio given the data
    mh_ratio <- (f(xstar) * g(x0, xstar)) / (f(x0) * g(xstar, x0))

    # Update the point (using function above)
    x0 <- find_xt(xstar, x0, mh_ratio)

    # Add the point to the path
    path[i] <- x0
  }
  return(path)
}

```

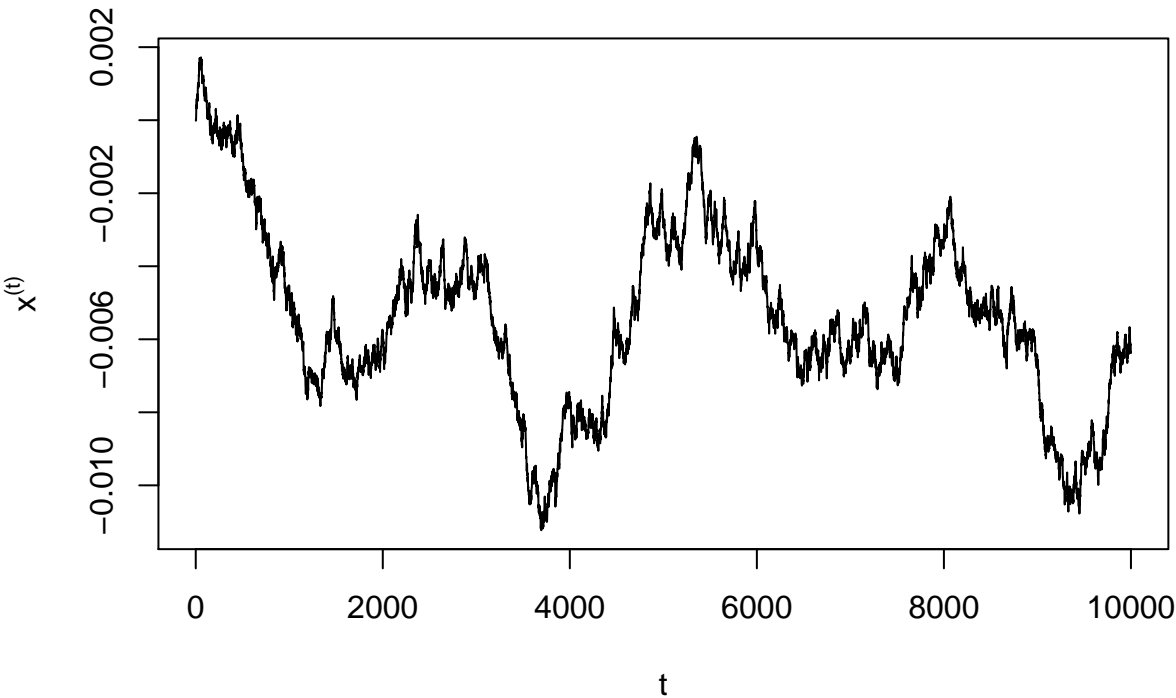
Next, we run the function using each of the starting points.

```

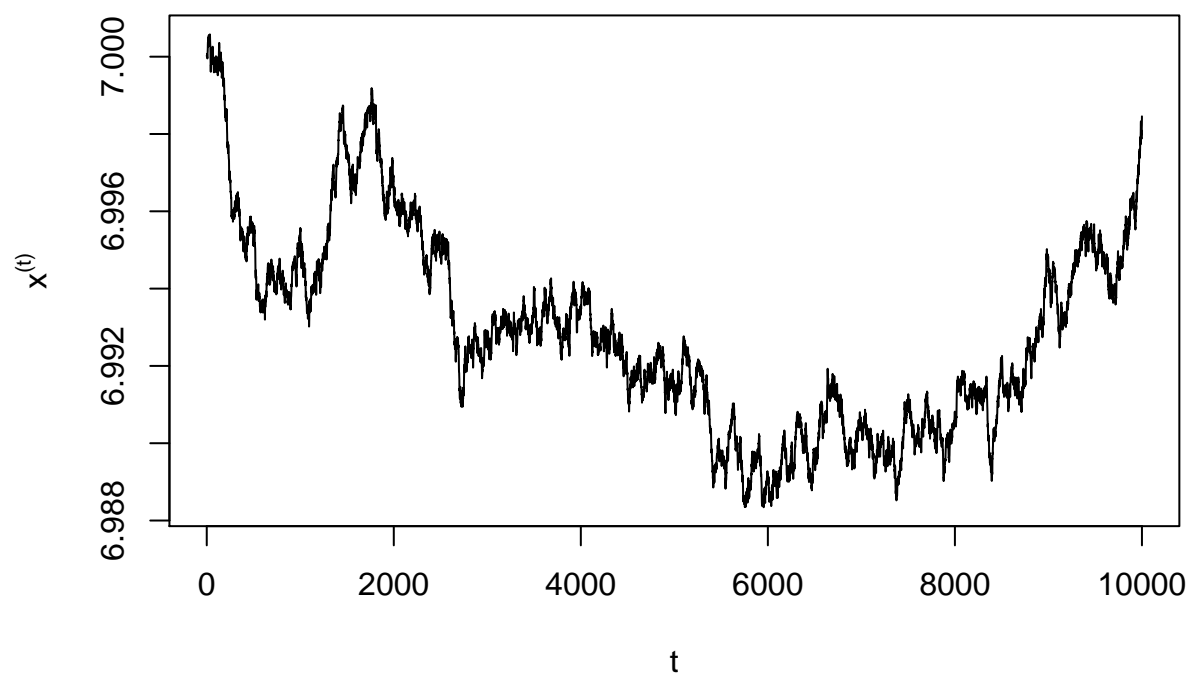
set.seed(73)
for(i in c(0, 7, 15)){
  path <- metropolis_hastings(g=g, rg=rg, f=f, x0=i, iterations=10000)
  plot(seq(1, 10000), path, 'l', main=paste0("Starting Point: ", i),
       xlab='t', ylab=TeX("$x^{\{t\}}$"))
}

```

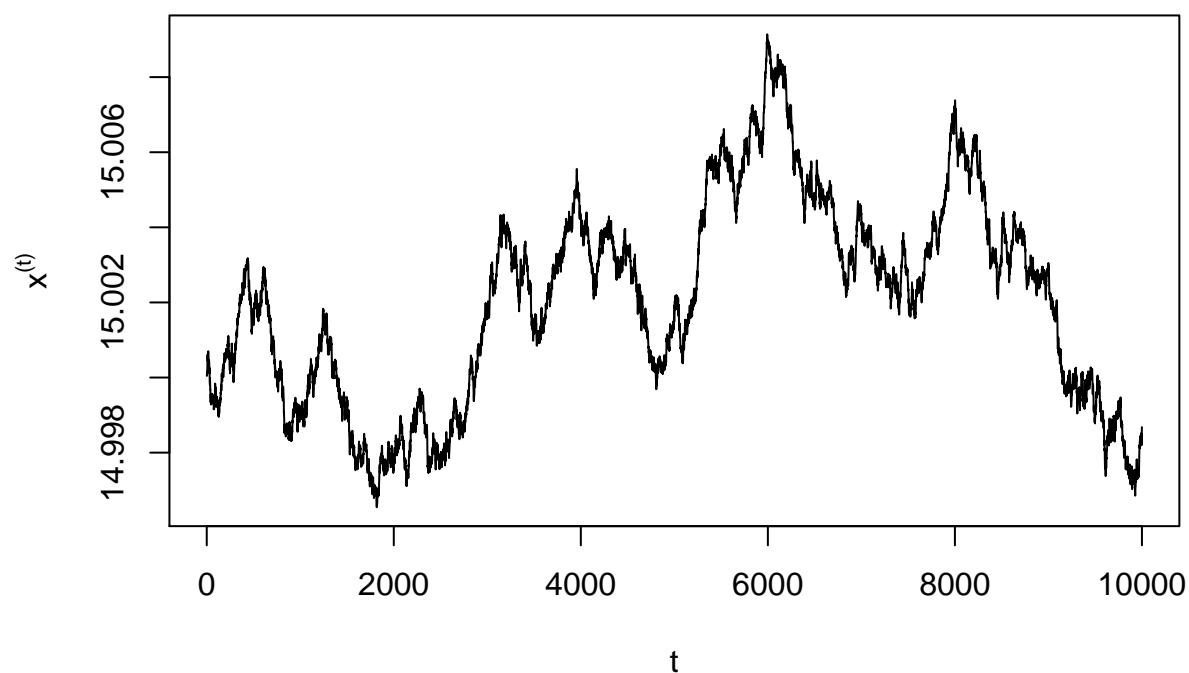
Starting Point: 0



Starting Point: 7



Starting Point: 15

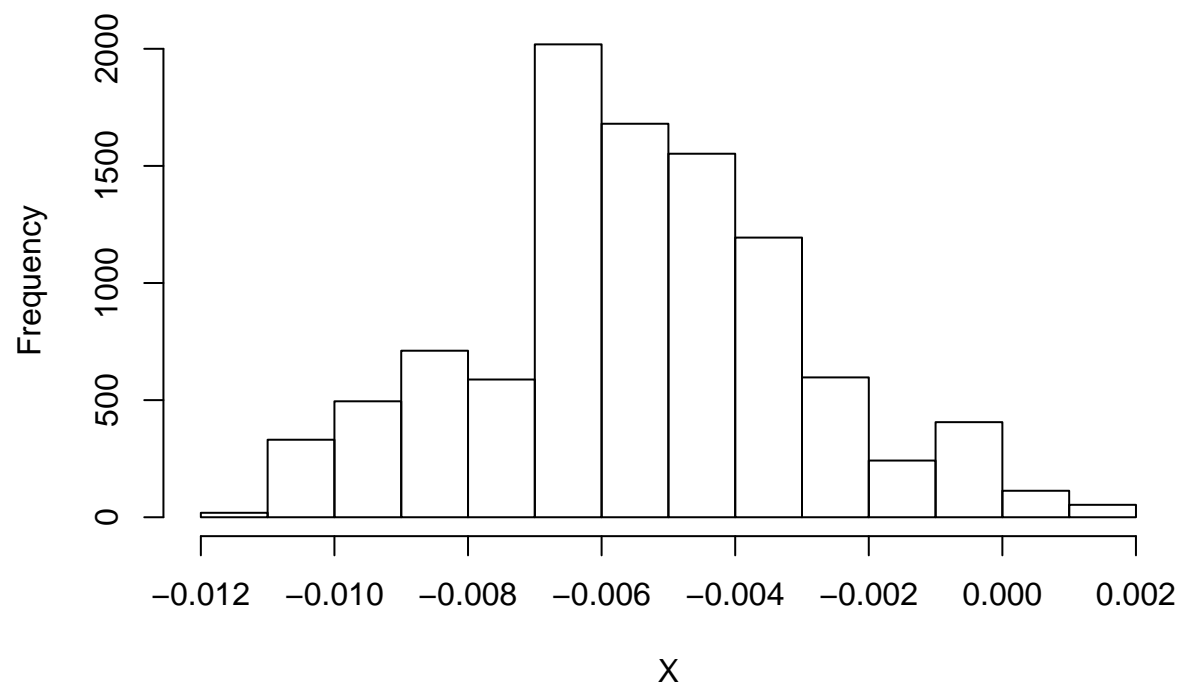


If only one of the given paths were available, it would be reasonable to reject the results, as the paths do not appear to have converged within some bounds. We might try to run more iterations to get a better sample.

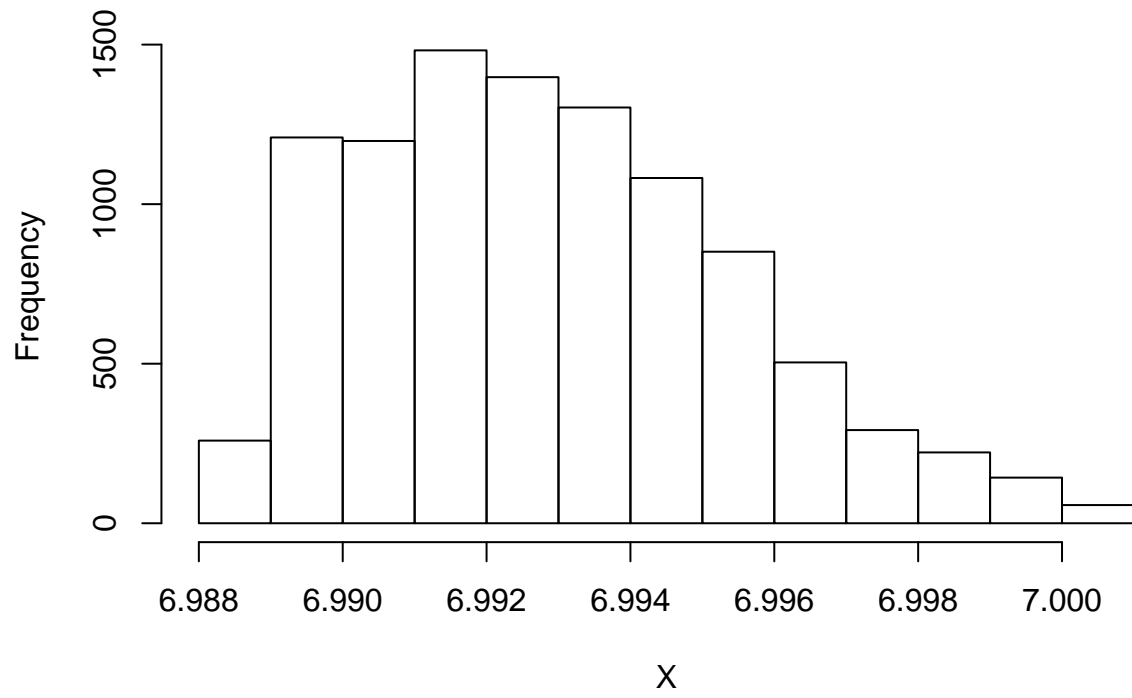
Next, we plot the histograms of each of the distributions.

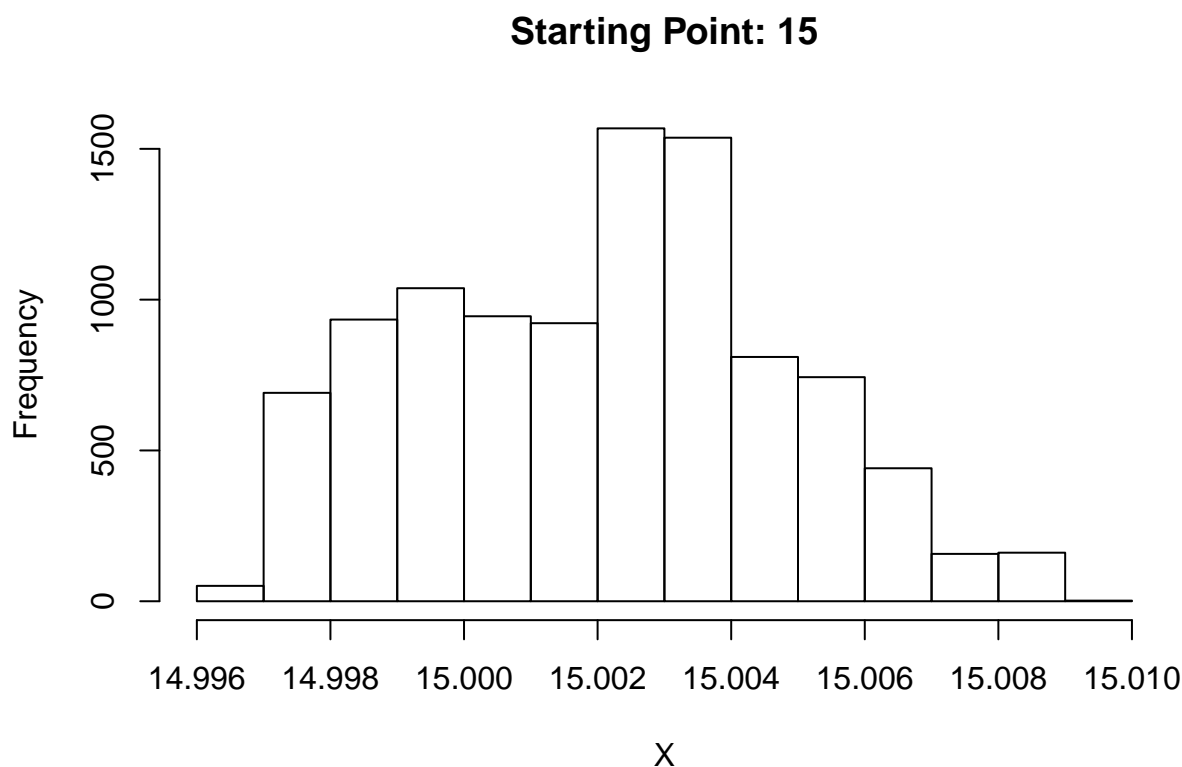
```
set.seed(73)
for(i in c(0, 7, 15)){
  path <- metropolis_hastings(g=g, rg=rg, f=f, x0=i, iterations=10000)
  hist(path, main=paste0("Starting Point: ", i), xlab='X')
}
```

Starting Point: 0



Starting Point: 7





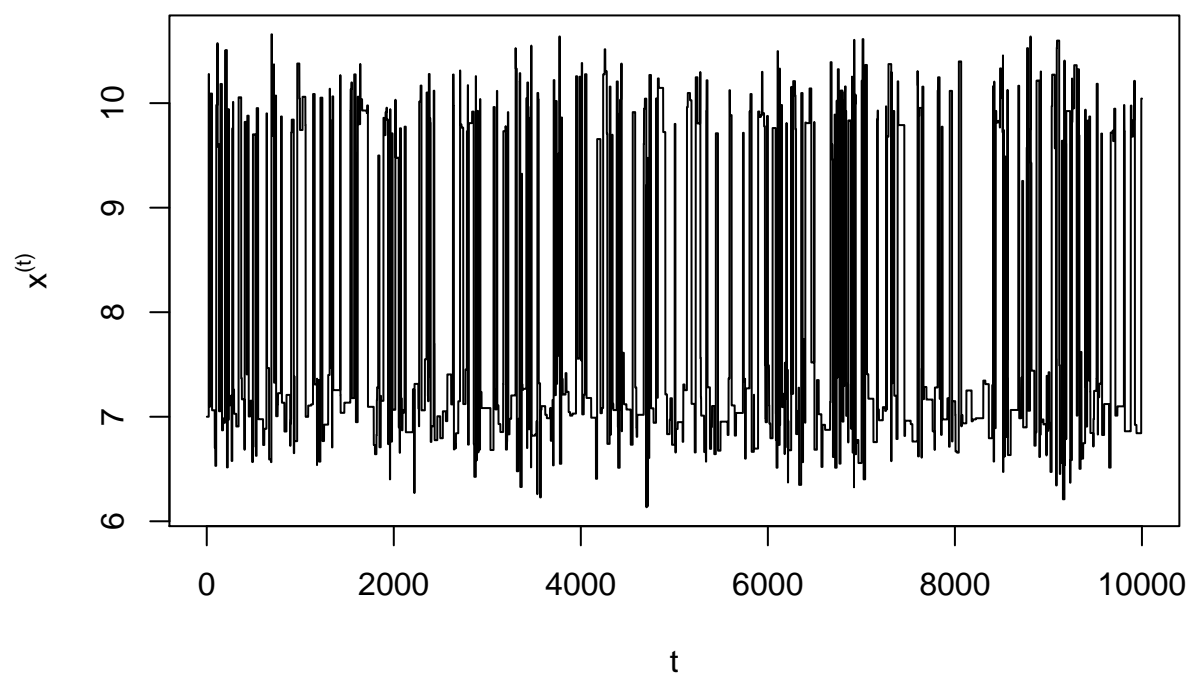
The histograms have density clustered around the starting value. Based on all of the paths, it's pretty clear that none of them represent the true density. Intuitively, it appears that increasing the variance on the proposal distribution may help, as the simulations are all clustered very tightly around the starting point, and don't appear to have the needed variance to sniff out the correct density.

Next, we change our proposal distribution to $U(0, 20)$ with $x_0 = 7$. We plot the path and the histogram.

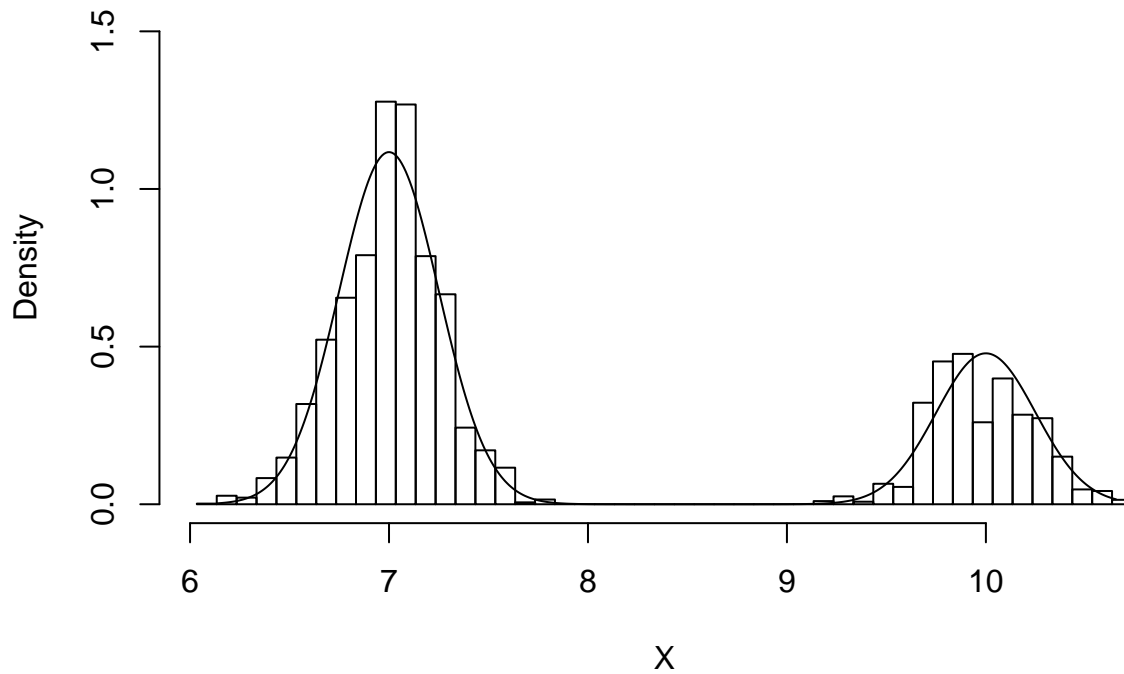
```
# Define proposal distributions
g <- function(xstar, x0) dunif(xstar, 0, 20)
rg <- function(x0) runif(n=1, min=0, max=20)

x0 <- c(7)
set.seed(73)
for(i in x0){
  path <- metropolis_hastings(g=g, rg=rg, f=f, x0=i, iterations=10000)
  plot(seq(1, 10000), path, 'l', main=paste0("Starting Point: ", i), xlab='t', ylab=TeX("$x^{\{t\}}$"))
  hist(path, breaks=seq(min(path) - .1, max(path) + .1, .1), main=paste0("Starting Point: ", i), xlab='X')
  lines(seq(min(path), max(path), .01), f(seq(min(path), max(path), .01)))
}
```

Starting Point: 7



Starting Point: 7



Using the uniform distribution given, our histogram looks correct, based on our definition for $f(x)$ (superimposed on the histogram). Also, in looking at the path given by the MH Algorithm, we note that this appears to be pretty clearly a mixture of two distributions with different means, as the path alternates between the two means with a very regular pattern.