Monte Carlo Methods HW 1

Kylie Taylor

2/4/2019

1. Use rejection sampling to sample from the density function $f(x) \propto (-logx)^2 x^3 (1-x)^2, 0 < x < 1$.

Carefully detail the method you use, and provide a figure of the histogram of the samples you obtained. What is the approximate, or actual if you can find it, probability of acceptance.

From in class notes, we know that the joint density function $f(x,u) = 1(u < h(x))g(x)$, where $f(x) = h(x)g(x)$ or $h(x) = \frac{f(x)}{g(x)}$. In order for rejection sampling to work we need to have $\frac{f(x)}{g(x)} \leq M < \infty$, meaning M is bounded. $f(x)$ is the desnsity we want to sample from and $g(x)$ is the one we can sample and take "candidate" samples from. The key is the marginal density of x from $f(x,u)$ is the target density $f(x)$, so if we can sample (x,u) from $f(x,u)$ then we can throw u away and keep the x. I am going to use a x coming from $g(x)$ and u coming from $U(0,1)$, then test $f(x,u) = M1(u < \frac{h(x)}{M})1(u < 1)g(x))$. If $u < \frac{h(x)}{M}$ then x comes from f, if not, sample again and again. We are interested in the probability of accepting $Pr(u < \frac{h(x)}{M})$. Using the law of total probability (conditioning on x) $Pr(u < \frac{h(x)}{M}) = \int Pr(u < \frac{h(x)}{M})g(x)dx = \frac{1}{M}\int h(x)g(x)dx$.

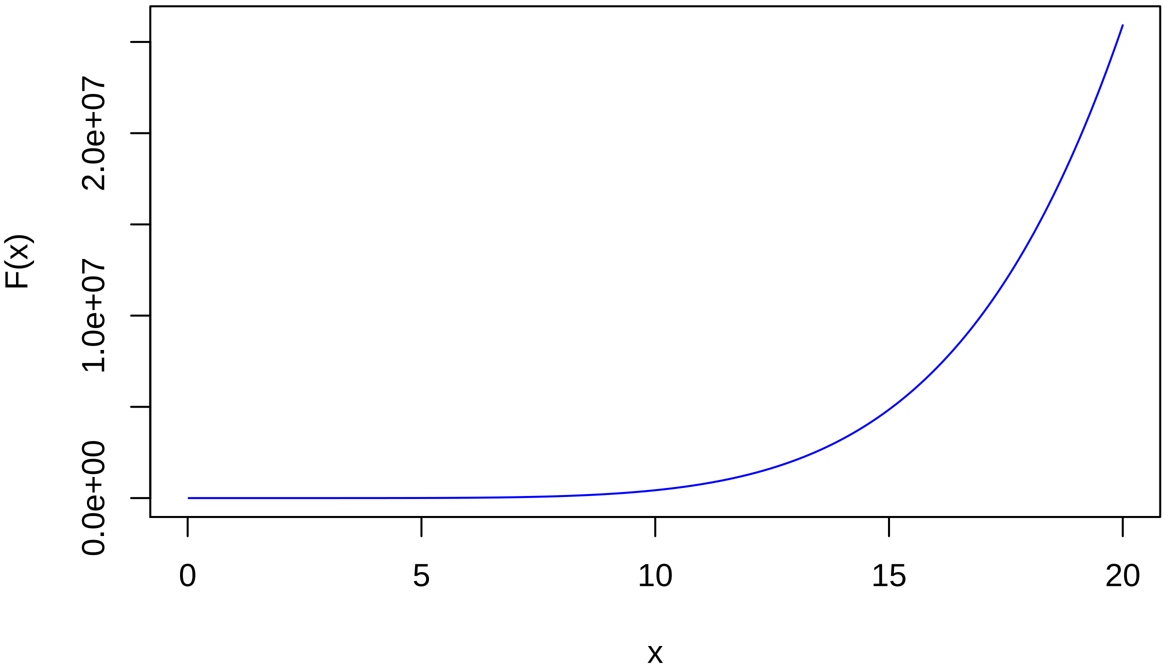
I will sample x's from a uniform density $g(x) = U(0,1) $, which makes $h(x) = \frac{f(x)}{g(x)} = (-logx)^2 x^3 (1-x)^2$.

First I will try to find the max, C, of $g(x)$ on the interval $0 < x < 1$. I did this by taking the maximum of the $log(g(x))$, since it should be the same as the max of $g(x)$.

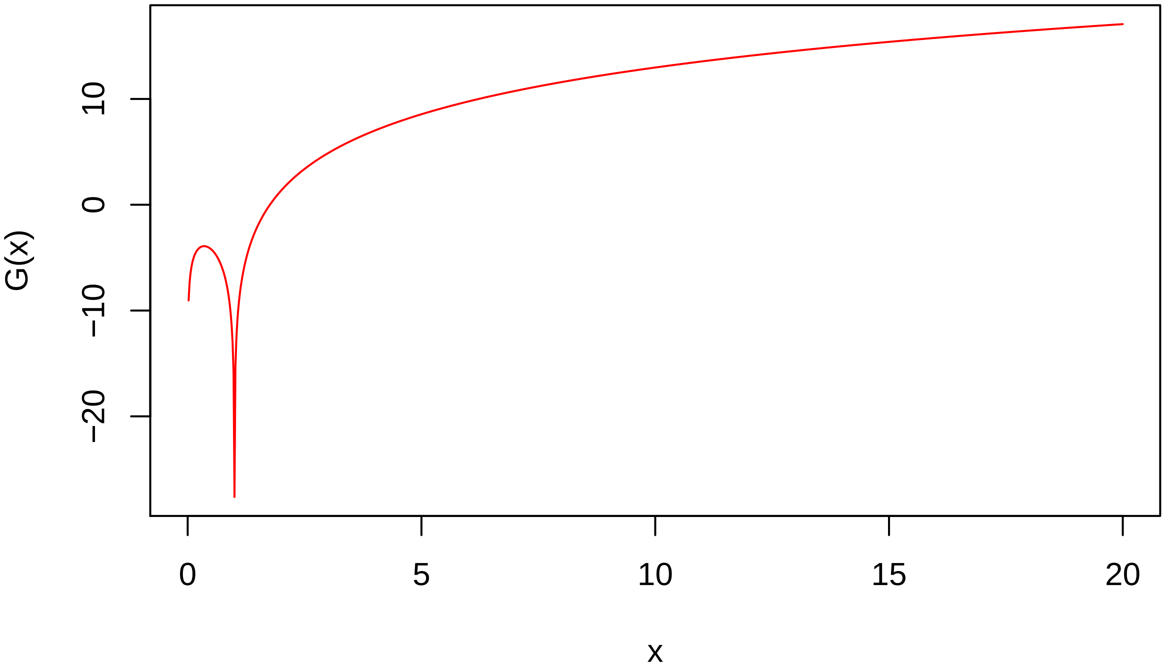
```r
F <- function(x){(-log(x))^2*x^3*(1-x)^2}
G <- function(x){log((-log(x))^2*x^3*(1-x)^2)}
optimize(G, c(0,1) , lower = 0.000001, upper = 0.99999,
         maximum = TRUE,
         tol = .Machine$double.eps^0.25)
```

```
## $maximum
## [1] 0.3517997
##
## $objective
## [1] -3.913744
```

```r
curve(F, 0, 20, n=1000, col='blue')
```



```r
curve(G, 0, 20, n=1000, col='red')
```

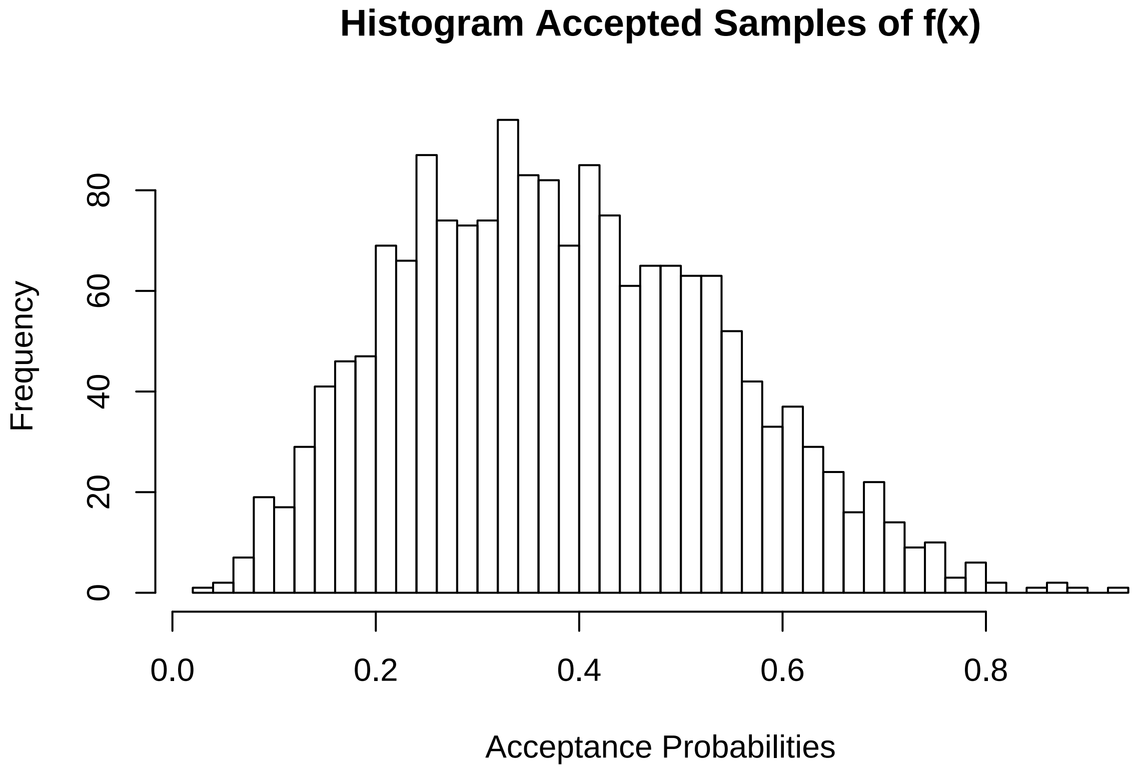


We find that the maximum value of $g(x)$ is when $x = 0.3517997$. We can use this to find that the normalizing constant, M, by $f(0.3517997) = (-log(0.3517997))^2 0.3517997^3(1 - 0.3517997)^2 = 0.0199656$. When both $h(x)$ and $g(x)$ integrate to 1, or are normalized, $\frac{1}{M}$ is the acceptance probability, but $\frac{1}{M} = 50.08615$, so $\frac{1}{M}$ is clearly not the acceptance probability. This is due to the fact that $h(x)$ does not integrate to 1 on the bounds $(-\infty, \infty)$.

I will use the following R code to provide a histogram of the samples I obtained and the probability acceptance.

As we can see, the average proability of accepting $f(x)$ is approximately 40%.

```
set.seed(1890)
rz1<-function (n)
{
        x <- runif(n, 0, 1) # randomly sample x from a uniform distribution
        u <- runif(n, 0, 1) # define u(x) as a uniform
        h <- (-log(x))^2*(x^3)*((1-x)^2) #define g(X) as this equation
        x[u < h] #if u is less than g(x) store as an x value
}

rz2<-function (n)
{
        x <- rz1(n) # store the number of times x was accepted
        len <- length(x) #count that number
        aprob <- len/n #divide that number by how many samples were taken
        shortby <- n - len  # number of x's that were rejectwed
        n2 <- round(shortby/aprob)
        x2 <- rz1(n2)
        x3 <- c(x, x2)
        x3
}

hist(rz2(1000),50, main = "Histogram Accepted Samples of f(x)", xlab = "Acceptance Probabilities")
```



```
mean(rz2(1000))
```

```
## [1] 0.3910113
```

2. Use Monte Carlo methods to evaluate the integral $I = \int_0^1 (-logx)^2 x^3 (1-x)^{5/2} dx$.

Describe in detail how you do this. Provide a graphical demonstration that your method has worked. If you fix the Monte Carlo sample size as N = 1000, is there a way to estimate the variance of your $\hat{I}_N$?

In problem #1, we showed that we are able to take samples from the function $h(x) = (-logx)^2 x^3 (1-x)^2$, in this problem we have $f(x) = (-logx)^2 x^3 (1-x)^{5/2}$. I will take samples of x from a $g(x) = U(0,1)$, since we know we are able to sample from a uniform density, making $h(x) = (-logx)^2 x^3 (1-x)^{5/2}$, remember $f(x) = h(x)g(x)$. This means we can represent I as $I = \int_0^1 f(x)dx = \int_0^1 h(x)g(x)dx$.

This leads to the definition of the Monte Carlo Integral: $\hat{I_n} = E(I_n|X_1, X_2, ..., X_n) = \frac{c}{n}\sum_{i=1}^{n} \frac{f(x_i)}{cg(x_i)} = \frac{1}{n}\sum_{i=1}^{n} \frac{f(x_i)}{g(x_i)}$, where c is the normalizing and $g_n(x) = cg(x)$, such that $c\int_0^1 g(x) = 1$.

The ideal $g(x)$ is one that minimizes the variance of the $\hat{I_n}$ and be able to sample from $g(x)$.

I will define $\bar{I} = \int_0^1 h(x)g_n(x)dx = \frac{\sum_1^N h(x_i)}{N}$ and $\hat{I} = \frac{\bar{I}}{c}$.

The steps of Monte Carlo Integration that I will follow are: 1) choose a pdf, $g(x)$ on [0,1]. 2) Generate data $X_1, X_2, ..., X_n$ from $g(x)$ 3) Estimate $\hat{I}$ by $\frac{1}{n}\sum_{i=1}^{n} \frac{f(x_i)}{g(x_i)}$

We can see that the Monte Carlo simulations estimated the integral of $f(x)$ well becuase when compared to the integral of $f(x)$, they both are approximately 0.0065. This reveals that my Monte Carlo estimate of the integral is accurate.

```r
set.seed(123456)
f <- function(x){(-log(x))^2*x^3*(1-x)^(5/2)} #Function to be integrated over [0,1].
g <- function(x){(-log(x))^2*x^3*(1-x)^2}
h <- function(x){(1 - x)^(1/2)}
n <- 1000

integrate(f, 0, 1)
```

```
## 0.006587444 with absolute error < 4.1e-07
```

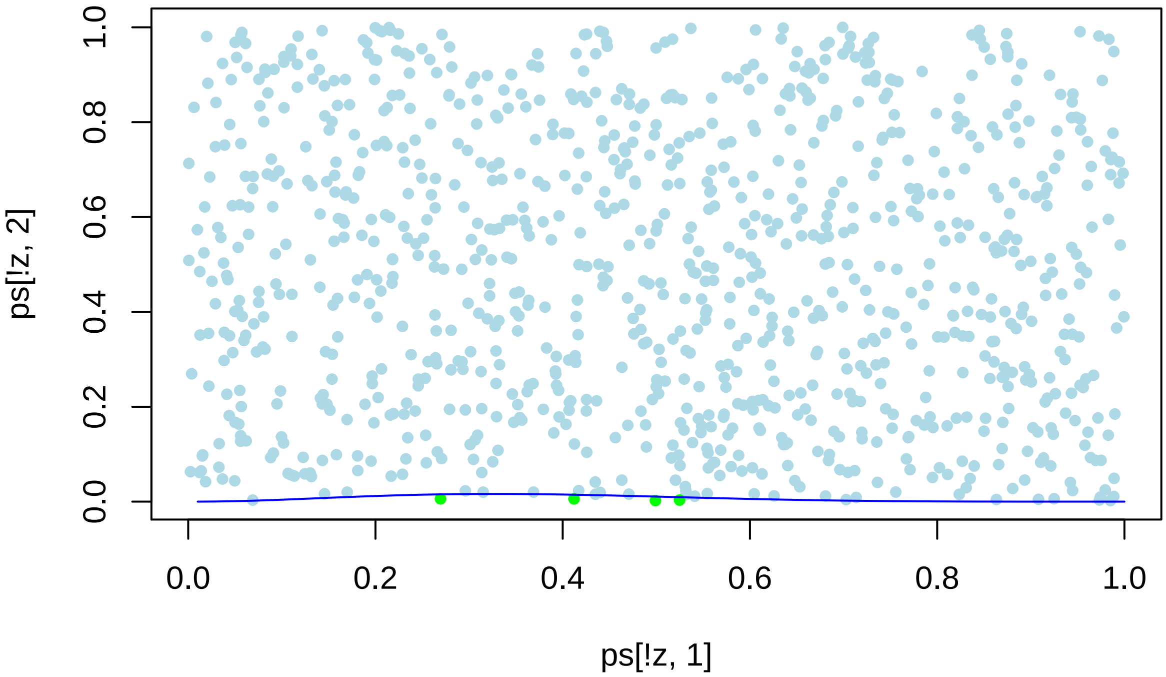
```r
MC.simple.est <-function(h, a, b, n){
  xi <- runif(n, a, b)# step 1
  mean <- mean(f(xi))      # step 2
  mean                    # step 3
}

MC.simple.est(h, 0, 1, n)
```

```
## [1] 0.006520177
```

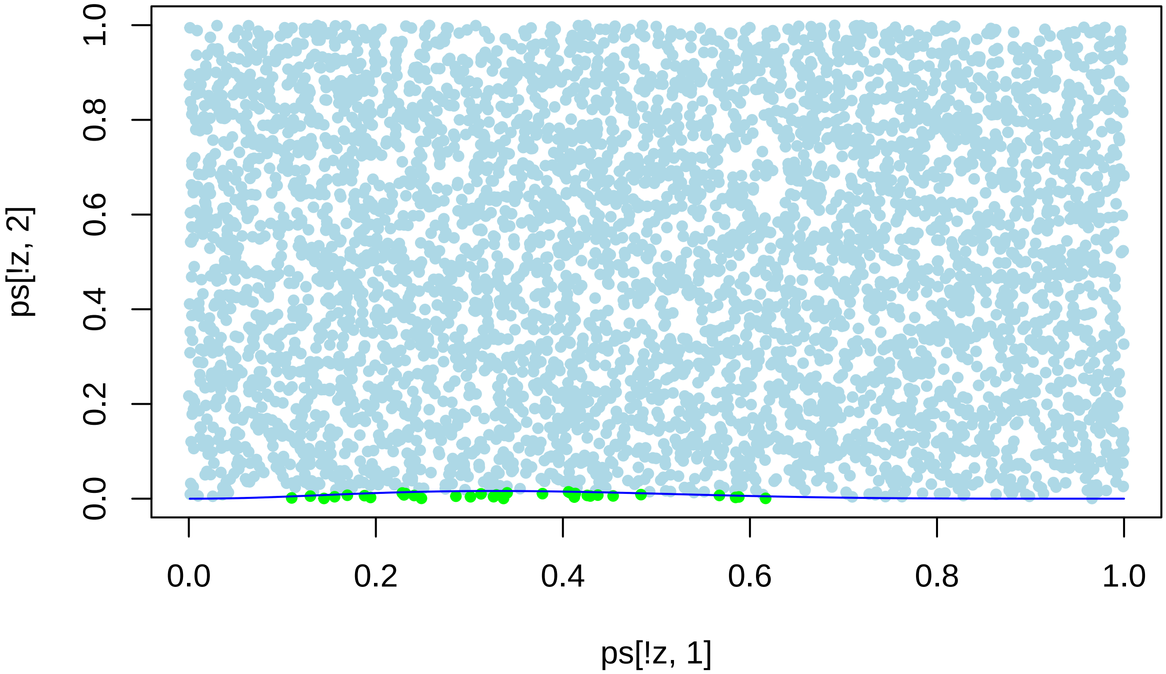
When I plot the $f(x)$ function and the MC simulations, we see that as I take more samples, the number of samples that I accept increases (intuitively). The last plot reveals that as I take more samples, the error of the acceptances decreases dramatically to the point where I am making nearly zero errors.

```r
n <- 1000

ps <- matrix(runif(2*n), ncol=2)
g <- function(x,y) y <= (-log(x))^2*x^3*(1-x)^(5/2)
z <- g(ps[,1], ps[,2])
plot(ps[!z,1], ps[!z,2], col='lightblue', pch=20)
```

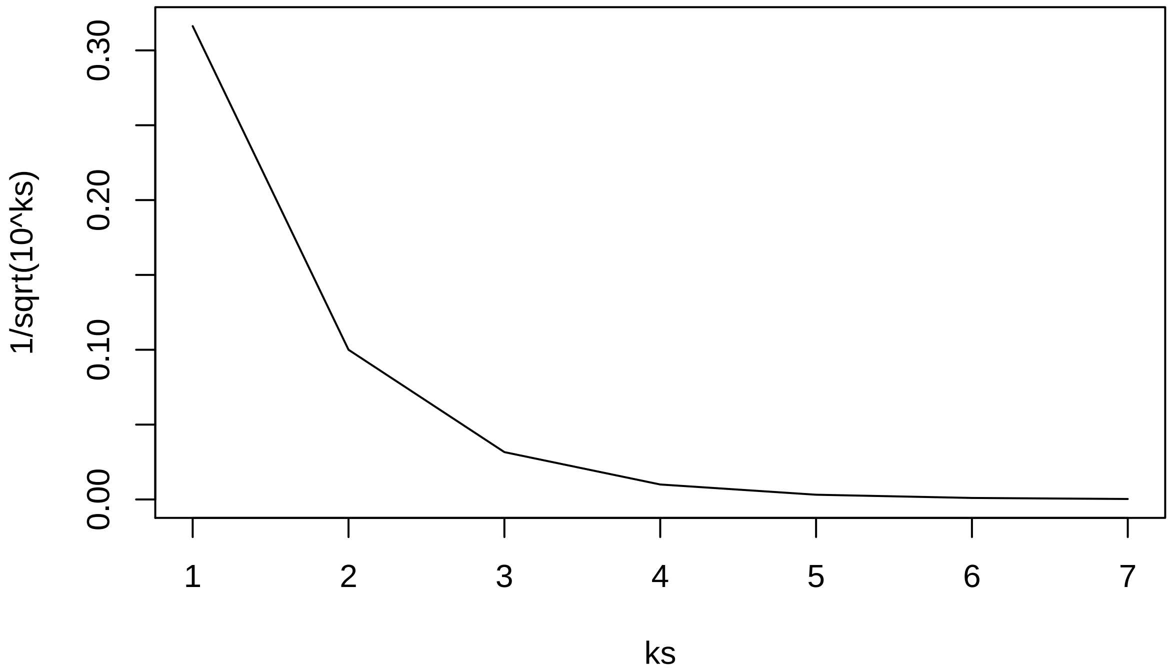
```
points(ps[z,1], ps[z,2], col='green', pch=20)
curve(f, 0,1, n=100, col='blue', add=TRUE)
```



```
ps <- matrix(runif(10*n), ncol=2)
g <- function(x,y) y <= (-log(x))^2*x^3*(1-x)^(5/2)
z <- g(ps[,1], ps[,2])
plot(ps[!z,1], ps[!z,2], col='lightblue', pch=20)
points(ps[z,1], ps[z,2], col='green', pch=20)
curve(f, 0,1, n=1000, col='blue', add=TRUE)
```



```r
ks <- 1:7
g <- function(k) {
  n <- 10^k
  f <- function(x,y) y <= (-log(x))^2*x^3*(1-x)^(5/2)
  z <- f(runif(n), runif(n))
  length(z[z]) / n
}

a <- sapply(ks,g)
plot(ks, 1/sqrt(10^ks), type='l')
```



There is a way to estimates the variance if I fix the sample size to n = 1000. The variance of this estimator is $Var(\hat{\Theta}) = Var(\frac{1}{n}\sum_{i=1}^{n} g(x_i)) = \frac{1}{n^2}Var(\sum_{i=1}^{n} g(x_i)) = \frac{Var(g(x))}{n}$. The variance of the Monte Carlo estimates is 3.628371e-05, nearly zero. This means that the Monte Carlo Simulations do a fair job in estimating the integral, validating my findings above.

```r
MC.var.est <- function(f, a, b, n) {
  xi <- runif(n, a, b)      # step 1
  var <- var(f(xi))    # step 2
  var            # step 3
}

MC.var.est(f, 0, 1, n)
```

```
## [1] 3.576548e-05
```

3. What is the acceptance probability when sampling a standard normal random variable with density $f(x) = \frac{1}{\sqrt{2\pi}} e^{\frac{-x^2}{2}}$ using a Cauchy density as proposal $h(x) = \frac{1}{\pi(1+x^2)}$ when using rejection sampling.

Verify this using simulation and plot a histogram of 1000 accepted samples.

This problem will follow the same steps as problem 1 since it is regards rejection sampling. In this case, instead of sampling x's from a uniform density, I will now sample x's from a Cauchy density.

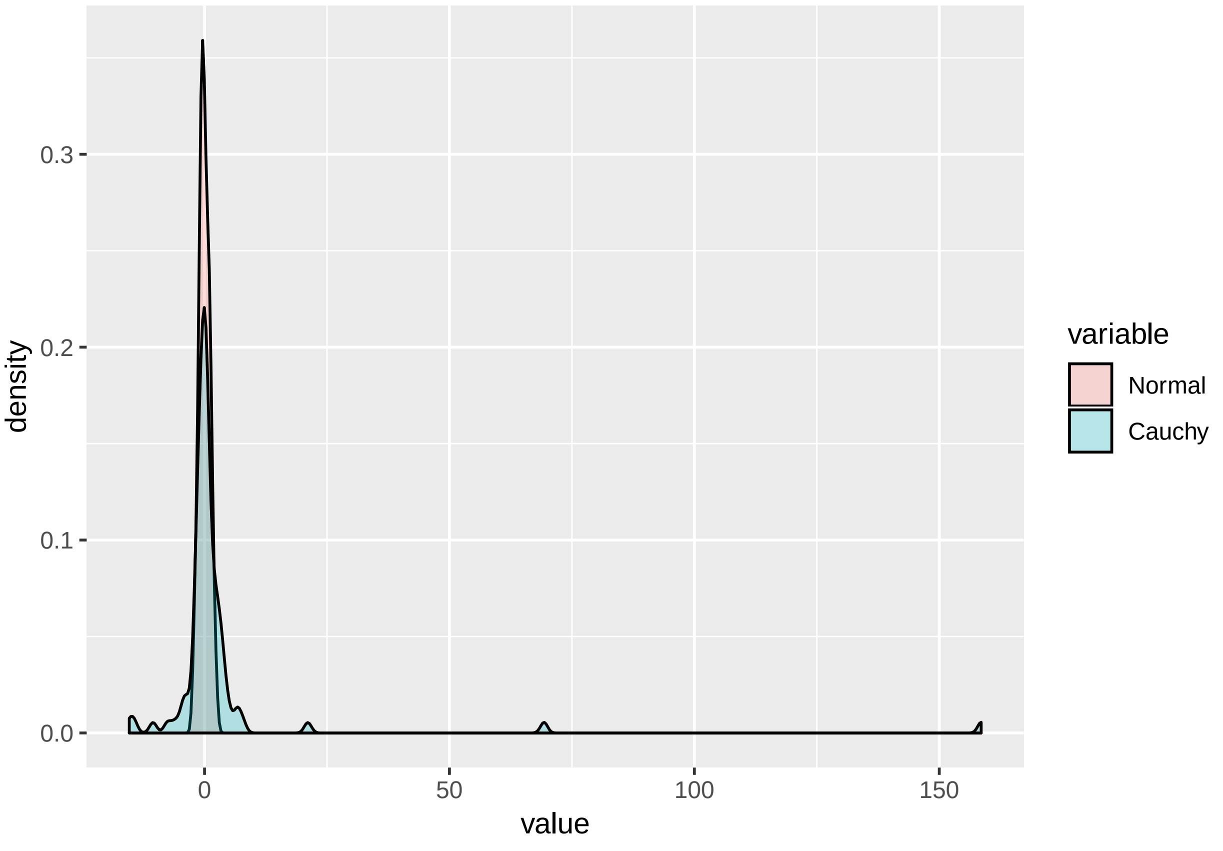
I will sample x's from a uniform density $h(x) = \frac{1}{\pi(1+x^2)}$, which makes $g(x) = \frac{f(x)}{h(x)} = \frac{\frac{1}{\sqrt{2\pi}} e^{\frac{-x^2}{2}}}{\frac{1}{\pi(1+x^2)}}$.

The calculated acceptance probability is approximately 0.7. This is clearly better than the 0.4 when sampling x from a unifrom density.

```r
set.seed(214554)
df <- data.frame(Normal=rnorm(100, 0, 1),Cauchy=rcauchy(100,0,1))
data<- melt(df)
```

```
## No id variables; using all as measure variables
```

```r
ggplot(data,aes(x=value, fill=variable)) + geom_density(alpha=0.25)
```



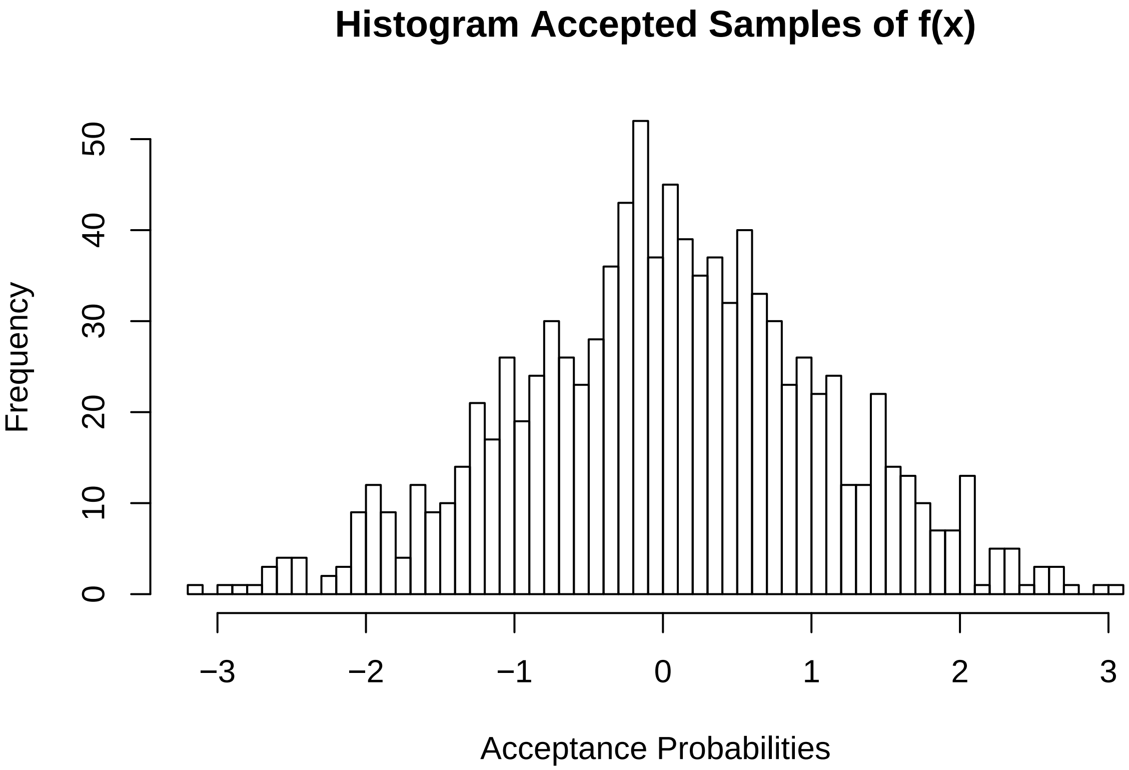
```r
f <- function(x) {exp(1)^(-(x^2)/2)/sqrt(2*pi)}
h <- function(x) {1/(pi(1+x^2))}
g <- function(x) {(pi*(1+(x^2))*exp(1)^(-(x^2)/2))/sqrt(2*pi)}

integrate(g, 0, 1)
```

```
## 1.384559 with absolute error < 1.5e-14
```

```
MC.simple.est <-function(h, a, b, n){
  xi <- runif(n, a, b)      # step 1
  g.mean <- mean(g(xi))      # step 2
  g.mean                    # step 3
}

MC.simple.est(g, 0, 1, n)
```

```
## [1] 1.383145
```

```
rz1<-function (n)
{
        x <- rcauchy(n, location = 0, scale = 1)
        u <- runif(n, 0, 1)
        g <- ((pi*(1+(x^2))*exp(1)^(-(x^2)/2))/sqrt(2*pi))
        x[u < g]
}

rz2<-function (n)
{
        x <- rz1(n)
        len <- length(x)
        aprob <- len/n
        shortby <- n - len
        n2 <- round(shortby/aprob)
        x2 <- rz1(n2)
        x3 <- c(x, x2)
        x3
}

hist(rz2(1000),50, main = "Histogram Accepted Samples of f(x)", xlab = "Acceptance Probabilities")
```



```
rz2<-function (n)
{
        x <- rz1(n)
        len <- length(x)
        aprob <- len/n
        shortby <- n - len
        n2 <- round(shortby/aprob)
        x2 <- rz1(n2)
        x3 <- c(x, x2)
        x3
        print(aprob)
}
```