Homework 9

Gustavo Esparza

05/04/2019

Libraries

```r
library(smoothmest) #for double exponential
library(mvtnorm)    #for MVN
library(ggplot2)
```

1

NOTE:

For this particular problem, there has been some ambiguity regarding the choice of $h(x)$. I have chosen to define $h(x) = \frac{1-e^{-1}}{1+x^2}$ so that $\int_0^1 h(x)f(x)$ is equivalent to the integral provided rather than having a constant value c left to be determined. The process is equivalent for both choices of $h(x)$.

Actual Value

```r
fun=function(x){return( (exp(-x))/(1+x^2))}
integrate(fun,0,1)
```

```
## 0.5247971 with absolute error < 5.8e-15
```

A

Let $f(x) = \frac{e^{-x}}{1-e^{-1}}, 0 \leq x \leq 1$. Initially, we are getting our X values from f(x). Then we have the following:

$$\theta = \int_0^1 h(x)f(x)dx = \int_0^1 \frac{e^{-x}}{1+x^2}dx, \text{ where } h(x) = \frac{1-e^{-1}}{1+x^2}$$

Now, we will get our X values from $g(x) = \frac{1}{\pi(1+x^2)}, -\infty \leq x \leq \infty$, the cauchy distribution.

First and foremost, we must define our standardized weights in the following manner. Let $w^*(x) = \frac{f(x)}{g(x)}$ and $w(x) = \frac{w^*(x)}{\sum_{i=1}^n w^*(x)}$

Then, we have our Importance Sampling estimate for the integral defined by

$$\theta_{IS} = \sum_{i=1}^n h(x_i)w(x_i)$$

It is important to note that the Cauchy dsitribution is defined for all real numbers, while our $f(x)$ truncated exponential is only defined in the interval $x \in [0, 1]$. Thus, we must remove any value that is outside of this domain and compute our estimate using the reduced estimate.

ESTIMATE

Therefore, we have the following calculations and result:

```r
f = function(x){
return(exp(-x)/(1-exp(-1)) )}

g = function(x){
  return(1/(pi*(1+x^2)))}

h = function(x){
  return((1-exp(-1))/(1+x^2))}

n = 20000

X = rcauchy(n,0,1)
X = X[which((X>0) & (X<1))]

n=length(X)

w_star = f(X)/g(X)
w = w_star/sum(w_star)

IS_est = sum(h(X)*w)

cat("Importance Sampling Estimate Using Cauchy Sampling =",IS_est,"")
```

```
## Importance Sampling Estimate Using Cauchy Sampling = 0.5240529
```

We can see that our estimate is quite close to the integral value found analytically.

Standard Error

The variance for our Importance Sampling estimate canbe defined as follows:

if $\bar{hw}$ is the Importance Sampling estimate found above, then

$$\sigma_{IS}^2 = \frac{1}{n*(n-1)} \sum_{i=1}^{n} (h(x)w(x) - \bar{hw})^2)$$

```r
#standard error
hwbar = IS_est
se_IS = (1/(n*(n-1)))*sum( ( h(X)*w - hwbar)^2)
se_IS = sqrt(se_IS)

cat("Importance Sampling Standard Error Using Cauchy Sampling =",se_IS,"")
```

```
## Importance Sampling Standard Error Using Cauchy Sampling = 0.0073657
```

We can see that the standard error for the estimate is quite small.

B

For B, we can repeat the steps used in A but now we will sample values from $g(x) = \frac{4}{\pi(1+x^2)}, 0 \leq x \leq 1$, the Truncated Cauchy Distribution. Since the new domain is already reduced, there is no need to get rid of any values. Here is the result for this $g(x)$:

ESTIMATE

```
f = function(x){
return(exp(-x)/(1-exp(-1)) )}

g = function(x){
  return(4/(pi*(1+x^2)))}

h = function(x){
  return((1-exp(-1))/(1+x^2))}

n = 20000

#using inverse function to get values in correct domain.
X=runif(20000,0,1)
X=tan((pi/4)*X)

#Weight
w_star = f(X)/g(X)
w = w_star/sum(w_star)

IS_est = sum(h(X)*w)

cat("Importance Sampling Estimate Using Truncated Cauchy Sampling =",IS_est,"")
```

```
## Importance Sampling Estimate Using Truncated Cauchy Sampling = 0.5249408
```

Then, for the truncated cauchy sampling, we can see that the estimate is still very close to the analytical value.

Standard Error

The variance for our Importance Sampling estimate can be defined as follows:

if $\bar{hw}$ is the Importance Sampling estimate found above, then

$$\sigma^2_{IS} = \frac{1}{n*(n-1)} \sum_{i=1}^{n} (h(x)w(x) - \bar{hw})^2)$$

```
#standard error
hwbar = IS_est
se_IS = (1/(n*(n-1)))*sum( ( h(X)*w - hwbar)^2)
se_IS = sqrt(se_IS)

cat("Importance Sampling Standard Error Using Truncated Cauchy Sampling =",se_IS,"")
```

```
## Importance Sampling Standard Error Using Truncated Cauchy Sampling = 0.003711799
```

The standard error for our estimate is still quite small, and in fact a bit smaller than the previous standrd error.

Thus, we can observe that both Cauchy sampling methods were effective at estimating the integral and reducing the respective variance. It is safe to say the cauchy distribution is effective for this Importance Sampling.

2

For the trivariate t distribution, $f(x, \Sigma) \propto (5 + x^T\Sigma^{-1}x)^{-4}$, we wish to approximate $\theta = P(-\infty \leq X_1 \leq 1, -\infty \leq X_2 \leq 4, -\infty \leq X_3 \leq 2)$. Then we have the following:

$\theta = \int_{-\infty}^{1}\int_{-\infty}^{4}\int_{-\infty}^{2} h(x)f(x)dx_1dx_2dx_3$, where $h(x) = 1_{[-\infty\leq X_i^1\leq 1, -\infty\leq X_i^2\leq 4, -\infty\leq X_i^3\leq 2]}$

Now, we will be generating random values from a trivariate normal distribution to approximate the desired probability, using the Importance Sampling Method.

Then if $X \sim g(x, \Sigma)$ is the Trivariate Normal Distribution, we can define the Standardized Sampling Weights as follows:

$$w(X_i) = \frac{w^*(X_i)}{\sum_{i=1}^{n} w^*(X_i)} = \frac{\frac{f(X_i,\Sigma)}{g(X_i,\Sigma)}}{\sum_{i=1}^{n}\frac{f(X_i,\Sigma)}{g(X_i,\Sigma)}}$$

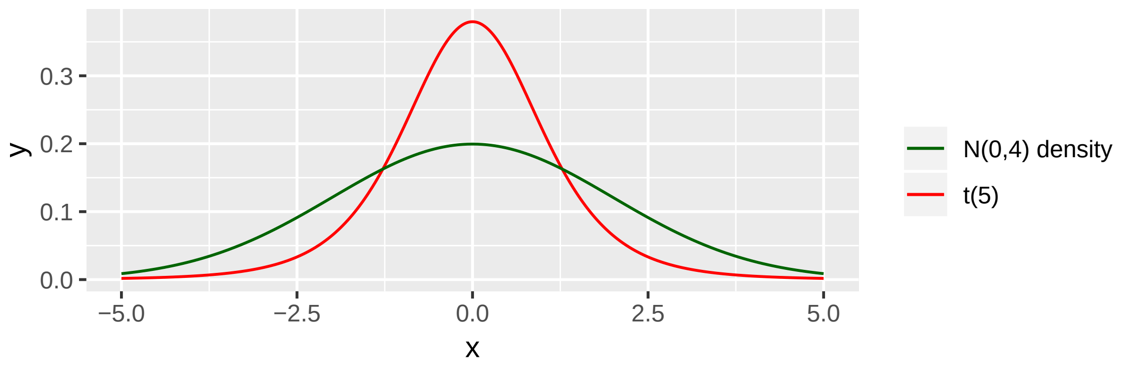
Then, provided $w(X_i)$, we can approximate θ with the following:

$$\hat{\theta}_{IS} = \sum_{i=1}^{n} h(X_i) * w(X_i)$$

FINDING APPROPRIATE TRIVARIATE NORMAL

Before implementing our Importance Sampling Method, we must first determine an approprtiate covariance for the normal distribution. We must first consider the univariate case where the variance of the Normal distribution envelopes the tails of the t-dsitribution. After a bit of investigation, it was found that a variance of 2 was suitable for the t-distribution privided. Here are the graphs that verify the envelope.

```r
#Creating Dataframe
x = seq(-5, 5, length.out = 1000)
dens_t = dt(x, df = 5)
dens_n = dnorm(x,0,2)

#Plot
df = data.frame(x, y1=dens_t, y2=dens_n)
ggplot( df,aes(x=x))+
geom_line(aes(y=y1, colour="t(5)"))+
geom_line(aes(y=y2, colour="N(0,4) density"))+ labs(y="y")+
scale_colour_manual("", values=c('dark green','red'))
```



We can see that the $N(0,4)$ distribution envelopes the tail ends of our $t(5)$ distribution, so this is valid for the variances of our trivariate normal distribution that we will be sampling from.

Thus, our covaraince matrix for the trivariate normal distribution can be defined as the multiple of the t-distribution covariance by the quantity of the normal variance, 4.

Now, we are able to implement our algroithm with a covariance matrix for the normal distribution that is appropriate for determining the probability for the t-distribution.

```
f = function(x, Sigma){
return((5+ t(x) %*% solve(Sigma) %*% x)^(-4)) }

g = function(x,mu,Sigma){
return(dmvnorm(x, mu, Sigma)) }

h = function(x){
if((x[1]<1) & (x[2]<4) & (x[3]<2))
  return(1)
else return(0) }

n = 20000
mu = c(0,0,0)

#Original Covariance
Sigma_t =matrix(c(1,3/5,1/3,3/5,1,11/15,1/3,11/15,1),3,3)

#Normal Covariance
Sigma = 4*Sigma_t

X = mvrnorm(n, mu, Sigma) #Sample from trivariate normal

#Initialize for running sum
w_star_sum=0
h_w_star_sum=0

for(i in 1:n){

h_x=h(X[i,]) #h

w_star = f(X[i,], Sigma_t)/g(X[i,],mu,Sigma) #w*

h_w_star = h(X[i,])*w_star #h * w* (numerator)

h_w_star_sum = h_w_star_sum + h_w_star #numerator sum

w_star_sum = w_star_sum + w_star # w* sum (demoniator)
}

theta_IS = h_w_star_sum/w_star_sum #total sum of h*w

print(theta_IS)
```

```
##              [,1]
## [1,] 0.7871509
```

Then, the estimate for our probability is very close to the approximate value provided in the text. Thus, sampling from the trivariate normal distribution with a covariance matrix that envelopes the tails of the t(5) distribution provides an accurate estimation for the desired probability.

3

A

Let $f(x) \sim N(0,1)$ and $g(x) \sim DoubleExp(\lambda)$. Then, for $\lambda > 0$, we will show that there exists $x > 0$ such that $f(x) = g(x)$. Then ,we have the following:

$$\frac{\lambda e^{-\lambda x}}{2} = \frac{e^{\frac{-x^2}{2}}}{\sqrt{2\pi}}$$
$$\implies e^{-\lambda x} = \frac{2}{\lambda\sqrt{2\pi}} e^{\frac{-x^2}{2}}$$
$$\implies -\lambda x = ln(\frac{2}{\lambda\sqrt{2\pi}}) - \frac{x^2}{2}$$
$$\implies \frac{-x^2}{2} + \lambda x + ln\left(\frac{2}{\lambda\sqrt{2\pi}}\right) = 0$$
$$\implies x^2 - 2\lambda x - 2ln\left(\sqrt{\frac{2}{\lambda^2\pi}}\right) = 0$$

Using the quadratic formula to solve for the values of x, we have the following result

$$x = \frac{2\lambda \pm \sqrt{4\lambda^2 + 8ln\left(\sqrt{\frac{2}{\lambda^2\pi}}\right)}}{2}$$

We can note, via graphing, that the quantity

$$4\lambda^2 + 8ln\left(\sqrt{\frac{2}{\lambda^2\pi}}\right) > 0$$

Then, we can assume that there are real solutions. So, we have found an $x > 0$ such that $f(x)$ and $g(x)$ intersect. Namely we have one solution being:

$$x = \frac{2\lambda + \sqrt{4\lambda^2 + 8ln\left(\sqrt{\frac{2}{\lambda^2\pi}}\right)}}{2}$$

Then, for any $\lambda > 0$, the double exponential and the standard normal density intersect at a value of $x > 0$. Since symmetry about the y-axis is assumed, we have also shown that the two densities intersect at a value of $x < 0$. Thus, we can be assured that the double exponential family does not dominate the standard normal density.

B

For $\lambda = 1$, we will find an optimal c that minimizes the number of rejections for an envelope $e(x) = cg(x)$. Then, we have the following:

$$c * \frac{e^{-x}}{2} \geq \frac{1}{\sqrt{2\pi}} e^{\frac{-x^2}{2}}$$

$$\implies c \geq \sqrt{\frac{2}{\pi}} e^{\frac{-x^2}{2}+x}$$

Now that we have found the c values that satisfy the envelope criteria, we must find the specific c that minimizes the number of rejections. Utilizing the log of c, we have the following

$$log(c) = \frac{1}{2} log(\frac{2}{\pi}) - \frac{x^2}{2} + x$$

$$\rightarrow \frac{\partial log(c)}{\partial x} = -x + 1 = 0 \rightarrow x^* = 1$$

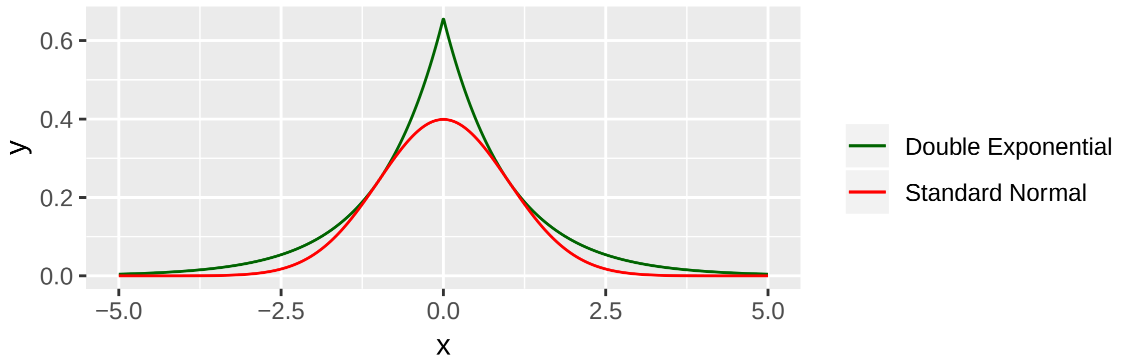
Now having the minimized x value, we have the optimial c that minimizes the number of rejections defined as

$$c^* = \sqrt{\frac{2}{\pi}} e^{\frac{1}{2}} \approx 1.315$$

Now that we have found the c value that minimizes the number of rejections, we will plot $f(x)$ and $g(x)$ with our new c value to ensure that we do in fact have an envelope of the function. Here is the graph needed to show this:

```r
f = function(x) {
  return(dnorm(x,0,1))}

g = function(x){
  return(ddoublex(x,0,1))}

c = sqrt(2/pi)*exp(1/2)

cg= function(x) {
  return(c*g(x))}
```

```r
df = data.frame(x,f(x),cg(x))

ggplot(df,aes(x=x))+
geom_line(aes(y=cg(x), colour="Double Exponential"))+
geom_line(aes(y=f(x), colour="Standard Normal"))+ labs(y="y")+
scale_colour_manual("", values=c('dark green','red'))
```



We can see that the Double Exponential does in fact envelope our Standard Normal distribution, thus our choice of c is justified.

C

CAUCHY ENVELOPE

In order to apply the ACCEPT/REJECT algorithm using a Cauchy Envelope, we must first find an optimal C that minimizes the number of rejections. Similar to the previous example, we have the following set-up:

$$c\frac{1}{1+x^2} \geq \frac{1}{\sqrt{2\pi}}e^{-\frac{x^2}{2}} \rightarrow c \geq \sqrt{\frac{\pi}{2}}e^{\frac{-x^2}{2}}(1+x^2)$$

Then, in order to minimize the number of rejections, we will once again consider the log of c for optimization:

$$log(c) = \frac{1}{2}log(\frac{\pi}{2}) - \frac{x^2}{2} + log(1+x^2)$$

$$\rightarrow \frac{\partial log(c)}{\partial x} = -x + \frac{2x}{1+x^2} = 0 \rightarrow 2x = x + x^3$$

$$\rightarrow x = x^3 \rightarrow x = 1 \textbf{ since x=1 maximizes our c instead of x=-1,0}$$

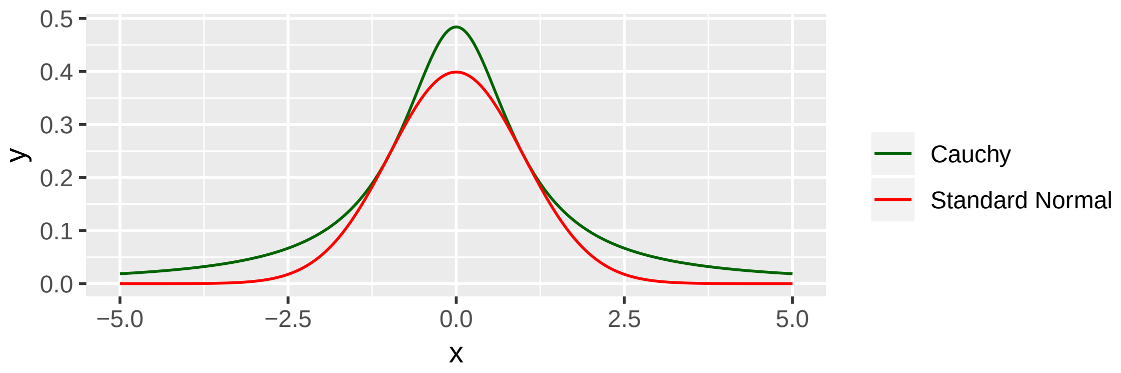
Then, using our minimized x value, we have the value of c that minimizes the number of rejections defined as

$$c^* = 2\sqrt{\frac{\pi}{2}}e^{\frac{-1}{2}} = \sqrt{2\pi} \times e^{\frac{-1}{2}} \approx 1.52$$

Now that we have found the c value that minimizes the number of rejections, we will plot $f(x)$ and $g(x)$ with our new c value to ensure that we do in fact have an envelope of the function. Here is the graph needed to show this:

```r
f = function(x) {
  return(dnorm(x,0,1))}

g = function(x){
  return(dcauchy(x,0,1))}

c= sqrt(2*pi)*exp(-1/2)

cg= function(x) {
  return(c*g(x))}
```

```r
df = data.frame(x,f(x), cg(x))

ggplot(df,aes(x=x))+
geom_line(aes(y=cg(x), colour="Cauchy"))+
geom_line(aes(y=f(x), colour="Standard Normal"))+ labs(y="y")+
scale_colour_manual("", values=c('dark green','red'))
```



We can see that the Cauchy does in fact envelope our Standard Normal distribution, thus our choice of c is justified.

Now, we are ready to implement the Accept/Reject Algorithm

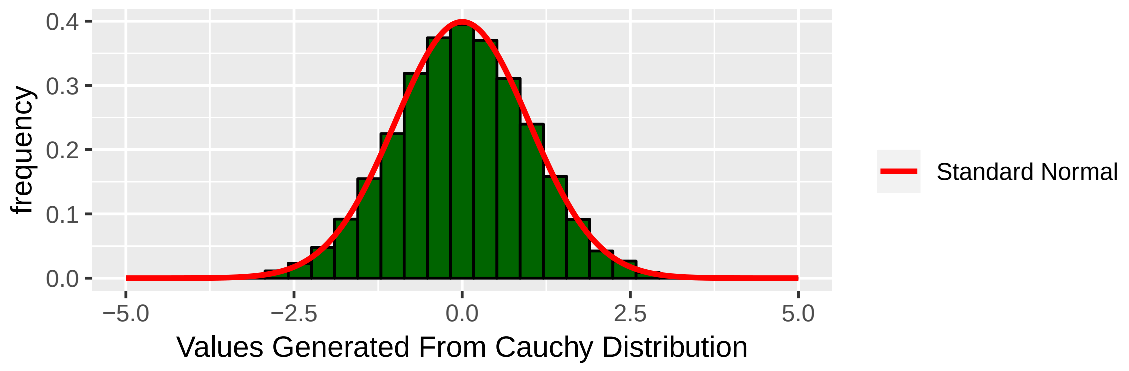
```r
#Accept/reject algorithm
n= 20000

X = rcauchy(n,0,1)

U = runif(n,0,1)

X= X[U<=f(X)/cg(X)]
```

```r
cat("Mean of generated X values from Cauchy RV = ",mean(X),
"\nStandard deviation of generated X values from Cauchy RV = ",sd(X),
"\nNumber of values accepted = ",length(X),"")
```

```
## Mean of generated X values from Cauchy RV =  0.002642633
## Standard deviation of generated X values from Cauchy RV =  1.005537
## Number of values accepted =  13159
```

Thus, we can see that our mean and standard deviation from our Accept/Reject Algorithm is very close to the Standard Normal values. Lastly, we will plot a histogram of our Accept/Reject data to see if it maintains a normal shape.

```r
df = data.frame(X)
dfnorm = data.frame(x,f(x))

ggplot(data=df, aes(x = X)) +
geom_histogram(aes(y=..density..), fill="dark green", color="black",bins = 30)+
  geom_line(data=dfnorm, aes(x=x,f(x), colour="Standard Normal"), lwd=1)+
  labs(x="Values Generated From Cauchy Distribution", y= 'frequency' )+
  scale_colour_manual("", values= 'red')
```



Thus, we can see that our Accept Reject algorithm does produce a normal distribution, as our normal distribution line encloses our histogram.

DOUBLE EXPONENTIAL ENVELOPE

For the Double Exponential Envelope, we will use the value of c provided in the first part of this problem.

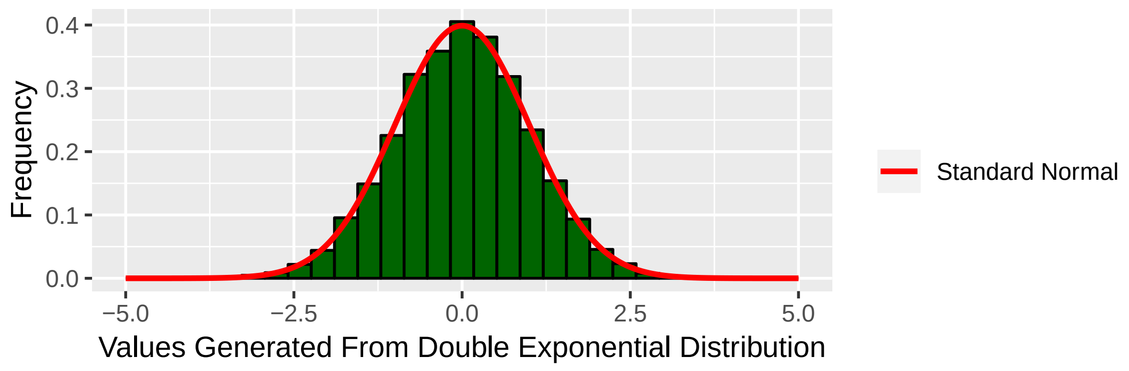
```
g = function(x){
  return(ddoublex(x,0,1))}

c = sqrt(2/pi)*exp(1/2)

cg = function(x){
  return(c*g(x))}

n= 20000
X= rdoublex(n,0,1)
U = runif(n,0,1)

X= X[U<=f(X)/cg(X)]
```

```
cat("Mean of generated X values from Double Exponential RV = ",mean(X),
"\nStandard deviation of generated X values from Double Exponential RV = ",sd(X),
"\nNumber of values accepted = ",length(X),"")
```

```
## Mean of generated X values from Double Exponential RV =  0.005576194
## Standard deviation of generated X values from Double Exponential RV =  0.9974349
## Number of values accepted =  15081
```

Again, we can see that our mean and standard deviation from our Accept/Reject Algorithm is very close to the Standard Normal values. Lastly, we will plot a histogram of our Accept/Reject data to see if it maintains a normal shape.

```
df = data.frame(X)
dfnorm = data.frame(x,f(x))

ggplot(data=df, aes(x = X)) +
geom_histogram(aes(y=..density..), fill="dark green", color="black",bins = 30)+
  geom_line(data=dfnorm, aes(x=x, f(x), colour="Standard Normal"), lwd=1)+
  labs(x="Values Generated From Double Exponential Distribution", y= 'Frequency' )+
  scale_colour_manual("", values= 'red')
```



Thus, we can see that our Accept Reject algorithm does produce a normal distribution, as our normal distribution line encloses our histogram.

COMPARISON

We have seen that both the Cauchy and Double Exponential envelopes did generate X values from the Standard Normal Distribution, as both sampling methods produced a distribution that had the desired Shape,mean and standard deviation.

When deciding which sampling distribution is preferred for the Accept/Reject method, we must consider which distribution accepted the most values when using the algorithm. In this case, the double exponential distribution produced more values than the cauchy distribution, thus I would recommend the Double Exponential Accept/Reject algorithm.