MA226 : Monte-Carlo Simulation
Generation of Random Numbers from Beta, Normal and Discrete Distributions

Assignment 5

Turkhade Hrushikesh Pramod
150123044

16-02-2017

1 Problem 1

We have to generate Random Numbers from Normal Distribution using Acceptance Rejection Method. For this, we take $g(x)$ as Double exponential function.

We can find c as follows:

$f(x)/g(x) = \sqrt{2/\pi}e^{(-x^2/2+|x|)} <= \sqrt{2e/\pi} = constant$

1.1 Source code of the solution

```
g<-function(x)
{
        return((1/2)*exp(-abs(x)))
}

gInv<-function(x)
{
        if(x<1/2)
        {
                return(log(2*x,base=exp(1)))
        }
        else
        {
                return(-log(2*(1-x),base=exp(1)))
        }
}

f<-function(x)
{
        return((1/sqrt(2*pi))*exp(-x*x/2))
}

n=5000
sample<-vector(length=n)
c<-sqrt(2*exp(1)/pi)
acc_count<-0
total_count<-0

i<-1

while(i<=n)
{
        x<-gInv(runif(1))

        if(c*g(x)*runif(1)<f(x))
        {
                sample[i]=x
                i<-i+1
                acc_count<-acc_count+1
        }
        total_count<-total_count+1
}
```

1.1.1 Histograms for the above data

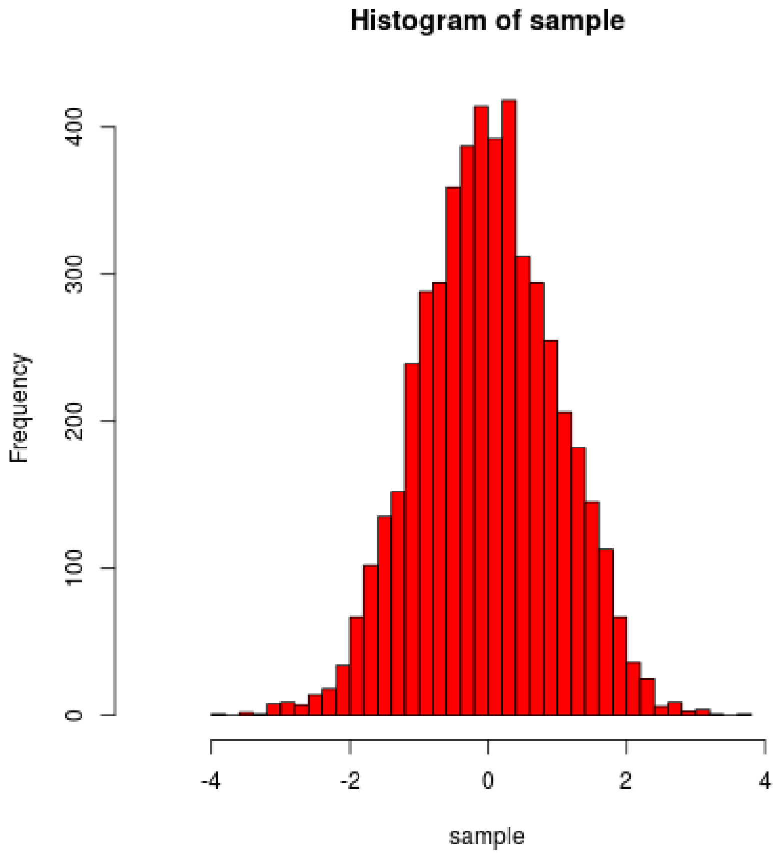

Abbildung 1: Histogram of Generated Normal Distribution

1.2 Analysis

As we can see from the graph, the histogram is quite similar to standard normal curve. The calculated values of various parameters are as follows:

$$Mean = 0.016 \tag{1}$$

$$Variance = 1.056 \tag{2}$$

$$SimulatedAcceptanceProbability = 0.755 \tag{3}$$

$$TheoriticalAcceptanceProbability = 0.760 \tag{4}$$

2 Problem 2

We have to generate Random Numbers from Half-Standard Normal Distribution using Acception Rejection Method. For this, we take $g(x)$ as Exponential distribution.

We can find c as follows:

$f(x)/g(x) <= \sqrt{2e/\pi} = constant$

2.1 Source code of the solution

```
g<-function(x,param)
{
        return(param*exp(-x*param))
}

gInv<-function(x,param)
{
        return(-log(1-x)/param)
}

f<-function(x)
{
        return((1/sqrt(2*pi))*exp(-x*x/2)*2)
}

n=5000
sample<-vector(length=n)
c<-sqrt(2*exp(1)/pi)
param<-1

acc_count<-0
total_count<-0

i<-1
while(i<=n)
{
        x<-gInv(runif(1),param)

        if(c*g(x,param)*runif(1)<f(x))
        {
                sample[i]=x
                i<-i+1
                acc_count<-acc_count+1
        }
        total_count<-total_count+1
}
```

2.1.1 Histograms for the above data

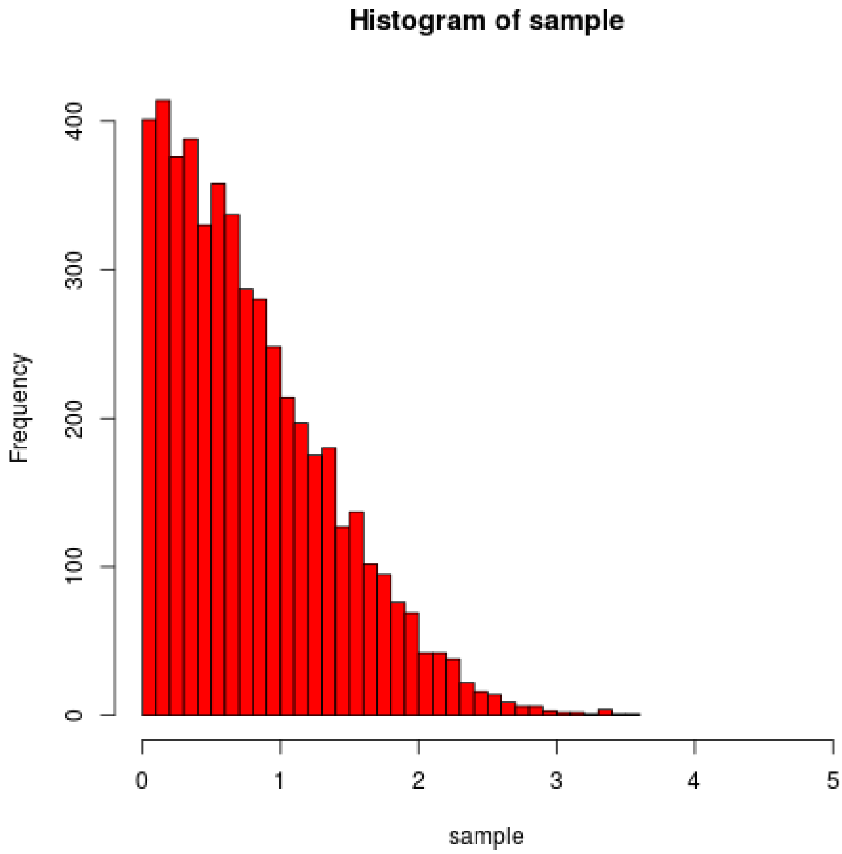

Abbildung 2: Histogram of Generated Half-Normal Distribution

2.2 Analysis

As we can see from the graph, the histogram is quite similar to half normal curve. The calculated values of various parameters are as follows:

$$Mean : 0.8041006 \tag{5}$$

$$Variance : 0.3709299 \tag{6}$$

3 Problem 3

We have to generate Random Numbers from Discrete Distribution using Acception Rejection Method and Inverse transform Method. The discrete distribution is as follows

Tabelle 1: Probabilities of Discrete Distribution

1	2	3	4	5
0.05	0.25	0.45	0.15	0.1

3.1 Source code of Inverse-Transform Method

```
n<-1000
dist<-c(0.05,0.25,0.45,0.15,0.1)
cdf<-c(0.05,0.25,0.45,0.15,0.1)
sample<-vector(length=n)

fInv<-function(x)
{
        i<-1
        while(cdf[i]<=x)
        {
                i<-i+1
                print(i)
        }
        return( i)
}

for(i in 2:5)
{
        cdf[i]<-cdf[i]+cdf[i-1]
}

for(i in 1:n)
{
        sample[i]=fInv(runif(1))
}
```

3.2 Source code of Acceptance Rejection Method

```
n<-10
dist<-c(0.05,0.25,0.45,0.15,0.1)
cdf<-c(0.05,0.25,0.45,0.15,0.1)
sample<-vector(length=n)

for(i in 2:5)
{
        cdf[i]<-cdf[i]+cdf[i-1]
}

i<-1
```

```
c<-0.45
lim<-5
while(i<=n)
{
	x<-lim*runif(1)
	x<-floor(x)+1
	px<-dist[x]

	if(c*runif(1)<px)
	{
		sample[i]=x
		i<-i+1
	}
}
```

3.3 Analysis

Following are the values of parameters in Acceptance Rejection Method. But, these are values will be inconsitant as we are generating only 10 numbers. Generatin more numbers will give us more insight on correctness of algorithm

$$Mean : 3.5 \tag{7}$$
$$Variance : 1.166667 \tag{8}$$

Following are the values of parameters in Inverse-Transform Method. By same reasoning above the values will be inconsitant in multiple runs of the code as number of samples are small.

$$Mean : 2.9 \tag{9}$$
$$Variance : 0.9888889 \tag{10}$$