MA 226
Monte Carlo Simulation
Assignment - 5

Amit Mittal
11012304
Department of Mathematics
I.I.T. Guwahati
Date:-07-02-2013

Assignments:-

1. ***Generate 1000 standard normal variates using standard Double - exponential distribution by acceptance-rejection method. Calculate the necessary constant c, where f (x) \g(x) ≤ c, f (x) and g(x) are the pdfs of standard normal and standard Double - exponential distribution respectively. Calculate the theoretical and simulated acceptance probability. How do you justify your generated random numbers are correct ?***
   Provide as many verification as you can.

2. ***Do the same exercise for generating random numbers from half standard normal distribution using exponential distribution with mean 1 by acceptance-rejection method.***

3. ***Consider the following discrete distribution.***

j	**1**	**2**	**3**	**4**	**5**
p_j	**0.05**	**0.25**	**0.45**	**0.15**	**0.10**

a) Generate 10 random numbers from the above probability mass function using usual procedure (inverse transform) of generating random number from discrete distribution defined on finite number of points. Calculate mean and variance of the generated numbers.

b) Generate 10 random numbers from the same probability mass function by acceptance - rejection principle. Calculate mean and variance of the generated numbers.

1.Solution:-
R code:

“

```
func_rand <- function(val, m){
	return(val/m);
}

func_val <- function(val, a, b, m){
	next_val <- ((a * val) + b) %% m;
	return(next_val);
}

func_f <- function(val){
	ans = exp(-0.5*((abs(val)-1)**2));
	return(ans);
}

func <- function(initial, a, b, m, end, initial_2){
	count<-0;
	total<-0;
	arr<-array(end+2);

	val <- func_val(initial, a, b, m);
	random <- func_rand(val, m);
	val_2 <- func_val(initial_2, a, b, m);
	random_2 <- func_rand(val_2, m);

	while(count<=end){
		if(random<=0.5){
			y = log(2*random);
		}
		else{
			y = -log(2-(2*random));
		}

		if(random_2<=func_f(y)){
			count = count+1;
			arr[count] = y;
		}

		val <- func_val(val, a, b, m);
```

```
        random <- func_rand(val, m);

        val_2 <- func_val(val_2, a, b, m);
        random_2 <- func_rand(val_2, m);
        total=total+1;
    }

    max<-0;
    min<-1;
    sum<-0;
    for(i in 1:end){
        cat(sprintf("arr[%g]\t%g\n", i, arr[i]));

        if(arr[i]>max){
            max = arr[i];
        }

        if(arr[i]<min){
            min = arr[i];
        }
        sum = sum + arr[i];
    }
    mean = sum/end;
    cat(sprintf("min = %g\nmax = %g\nmean = %g\nPractical Acceptance
Probability = %g\n", min, max, mean, 1000/total));
    theory_probab = 1/sqrt(2*exp(1)/pi);
    cat(sprintf("Theoritical Acceptance Probability = %g\n", theory_probab));

    hist(arr, 100,freq = TRUE, include.lowest = TRUE, right = TRUE, density =
NULL,
        angle = 45, col = '#1E90FF', border = NULL,
        main = paste("Histogram of frequencies and total values
        generated=",end), xlab = "Random Values" ,
        axes = TRUE, plot = TRUE, labels = FALSE);
}

init_value <- 3452;
init_value_2 <- 5691;
bound <- 1000;
func(init_value, 16807, 0, 2**31-1, bound, init_value_2);
”
```

Calculation **:**

We know,

$$c = f(x)/g(x)$$

Now,

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

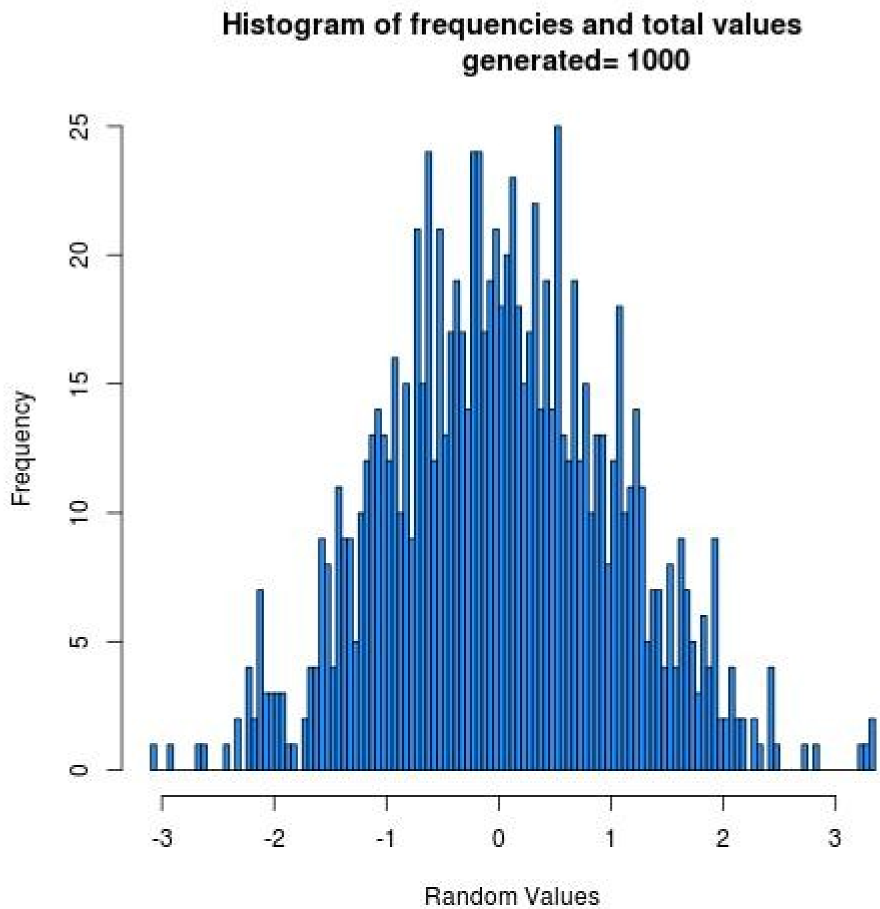
So,

$$f(x)/g(x) = \sqrt{2/\pi}\,e^{-x^2+|x|} \leqslant \sqrt{2e/\pi} = 1.31548924696$$

So, c = 1.31548924696 (theoretical)

So, acceptance probability = $1/c$ = 0.7601735

Histograms :


Histogram of frequencies and total values
generated= 1000
Frequency
0
5
10
15
20
25
-3
-2
-1
0
1
2
3
Random Values

<u>Observation</u> :

1) mean = 0.0237833
2) variance = 1.03443
3) acceptance probability = 0.76746 (practically)
4) acceptance probability = 0.7601735 (theoritically)
5) percentage error in acceptance probability simulated = 0.95853 %

<u>Justification</u> :

1)The mean is very close to 0 , the theoretical mean.
2)The variance is very close to 1 , the theoretical variance.
3)Percentage error in variance simulated = 3.443 %
4)The acceptance probability is very close to 0.7601735 , the theoretical acceptance probability .
5)Percentage error in acceptance probability simulated = 0.95853 %

<u>Result</u> :

1)We have successfully generated 1000 standard normal variates using Double-exponential distribution.
2)From the justifications above, we can say that generated random numbers are correct.

2.Solution:-

R code:

“

```
func_rand <- function(val, m){
	return(val/m);
}

func_val <- function(val, a, b, m){
	next_val <- ((a * val) + b) %% m;
	return(next_val);
}

func_f <- function(val){
	ans = exp(-0.5*((abs(val)-1)**2));
	return(ans);
}

func <- function(initial, a, b, m, end, initial_2){
	count<-0;
	total<-0;
	arr<-array(end+2);

	val <- func_val(initial, a, b, m);
	random <- func_rand(val, m);
	val_2 <- func_val(initial_2, a, b, m);
	random_2 <- func_rand(val_2, m);

	while(count<=end){
		y = -log(1-random);

		if(random_2<=func_f(y)){
			count = count+1;
			arr[count] = y;
		}

		val <- func_val(val, a, b, m);
		random <- func_rand(val, m);

		val_2 <- func_val(val_2, a, b, m);
		random_2 <- func_rand(val_2, m);
		total=total+1;
```

```
	}

	max<-0;
	min<-1;
	sum<-0;
	for(i in 1:end){
		cat(sprintf("arr[%g]\t%g\n", i, arr[i]));

		if(arr[i]>max){
			max = arr[i];
		}

		if(arr[i]<min){
			min = arr[i];
		}
		sum = sum + arr[i];
	}
	mean = sum/end;
	cat(sprintf("min = %g\nmax = %g\nmean = %g\nPractical Acceptance
Probability = %g\n", min, max, mean, 1000/total));
	theory_probab = 1/sqrt(2*exp(1)/pi);
	cat(sprintf("Theoritical Acceptance Probability = %g\n", theory_probab));

	hist(arr, 100,freq = TRUE, include.lowest = TRUE, right = TRUE, density =
NULL,
		angle = 45, col = '#1E90FF', border = NULL,
		main = paste("Histogram of frequencies and total values
		generated=",end), xlab = "Random Values" ,
		axes = TRUE, plot = TRUE, labels = FALSE);

	dev.copy(jpeg,'img.jpg');
	dev.off();
}

init_value <- 113;
init_value_2 <- 1031;
bound <- 1000;

func(init_value, 16807, 0, 2**31-1, bound, init_value_2);
”
```

Calculation **:**

We know,

$$c = f(x)/g(x)$$

Now,

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

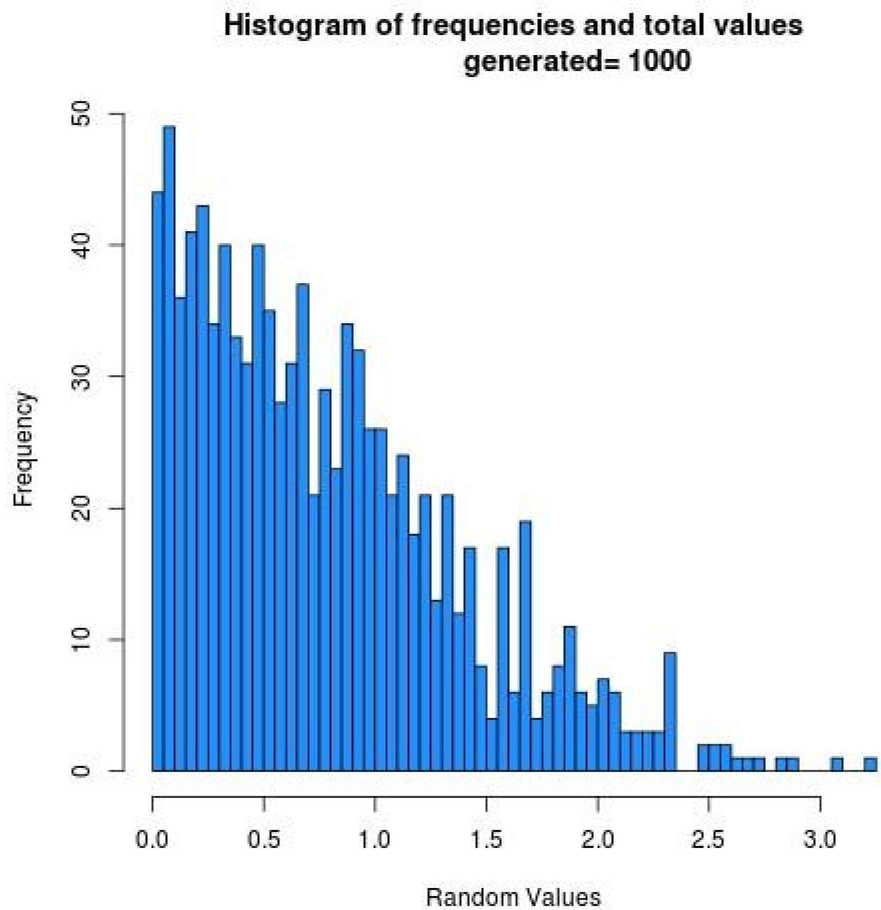
So,

$$f(x)/g(x) = \sqrt{2/\pi}\,e^{-x^2+x} \leqslant \sqrt{2e/\pi} = 1.31548924696$$

So, c = 1.31548924696 (theoretical)

So, acceptance probability = $1/c$ = 0.7601735

Histograms **:**



Histogram of frequencies and total values
generated= 1000
Frequency
Random Values
0
10
20
30
40
50
0.0
0.5
1.0
1.5
2.0
2.5
3.0

Observation ***:***

1) mean = 0.789702
2) variance = 0.362459
3) acceptance probability = 0.768049 (simulated)
4) acceptance probability = 0.7601735 (calculated)
5) percentage error in acceptance probability simulated = 1.03601 %

Justification :

1)The mean is very close to 0.7978845608 , the theoretical mean.
2)The variance is very close to 0.3633802276 , the theoretical variance.
3)Percentage error in mean simulated = 1.025531913 %
4)Percentage error in variance simulated = 0.25351616 %
5)The acceptance probability is very close to 0.7601735 , the theoretical acceptance probability .
6)Percentage error in acceptance probability simulated = 1.03601 %

Result :

1)We have successfully generated 1000 standard half normal variates using exponential distribution with parameter 1.
2)From the justifications above, we can say that generated random numbers are correct.

3.Solution:-

Part (a)

R code:

“

```
generateDIS<-function(num){

a<-40692
m<-2147483399
x0<-11
countnum<-0
countfreq<-numeric(5)
arraysummary<-array()

y<-(a*x0)%%m
u<-y/m
p<-c(0.05, 0.25, 0.45, 0.15, 0.10)

        while(countnum < num){

                prob<-0
                index<-1
                while(index<=5){

                        prob<-prob+p[index]
                        if(u<prob)
                                break

                        index<-index+1
                }

                #print(index)
                countnum<-countnum+1
                countfreq[index]<-countfreq[index]+1
                arraysummary[countnum]<-index
                y<-(a*y)%%m
                u<-y/m
        }
        print(summary(arraysummary))
```

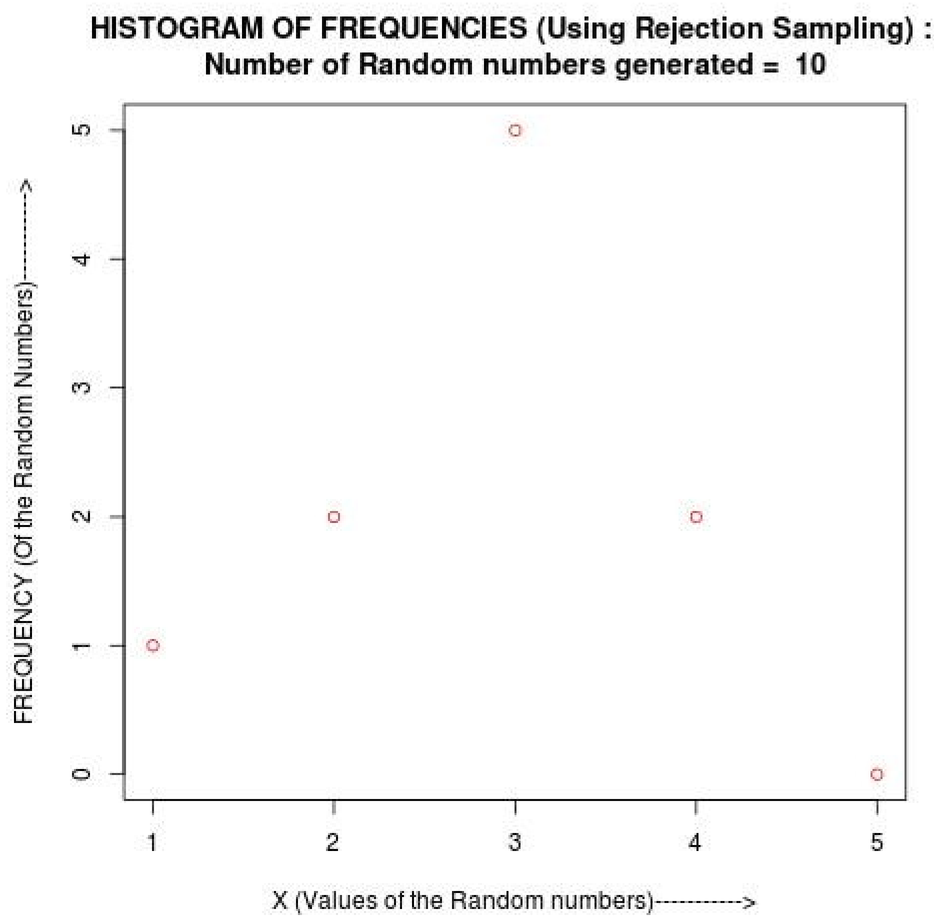
```
	print(countfreq)
	plot(countfreq, xlab = "X (Values of the Random numbers)----------->", ylab = 
"FREQUENCY (Of the Random Numbers)----------->", col = rgb(1,0,0,1), main = 
paste("HISTOGRAM OF FREQUENCIES (Using Rejection Sampling) : \nNumber of 
Random numbers generated = ", num))
	dev.copy(jpeg, "Q3a.jpeg")
	dev.off()
}

generateDIS(10)
```

”

Mean and Variance :

1) Simulated mean is 2.80
2) Calculated mean is 3
3) Calculated variance is 1

(This deviation occurs because only 10 numbers are generated)



Part (b)

R code:

“

```
generateDISACCREJ<-function(num){

a<-c(16807, 40692)
m<-c(((2^31)-1), 2147483399)
x0<-c(5,5)
u<-array(2)
y<-array(2)
countnum<-0
countaccept<-0
arraysummary<-array()
q = c(0.2, 0.2, 0.2, 0.2, 0.2)
p = c(0.05, 0.25, 0.45, 0.15, 0.10)
countfreq<-numeric(5)
M<-2.25

y[1]<-(a[1]*x0[1])%%m[1]
y[2]<-(a[2]*x0[2])%%m[2]
u[1]<-y[1]/m[1]
u[2]<-y[2]/m[2]

        while(countaccept < num){

                index<-1
                prob<-0

                while(index <= 5){

                        prob<-prob+q[index]
                        if(u[1] < prob)
                                break

                        index<-index+1
                }

                if(u[2]<=(p[index]/(M*q[index]))){
                        #print(paste("Accepted : ", index), quote = FALSE)
                        countaccept<-countaccept+1
```

```
                countfreq[index]<-countfreq[index]+1
                arraysummary[countnum]<-index
            }

            countnum<-countnum+1
            y[1]<-(a[1]*y[1])%%m[1]
            y[2]<-(a[2]*y[2])%%m[2]
            u[1]<-y[1]/m[1]
            u[2]<-y[2]/m[2]
        }
        print(paste("Number of values generated for testing : ", countnum), quote =
FALSE)
        print(paste("Number of values accepted : ", countaccept), quote = FALSE)
        print(paste("Probability of Acceptance (Experimental) : ",
countaccept/countnum), quote = FALSE)
        print(paste("Probability of Acceptance (Theoretical) : ", 1/M), quote = FALSE)
        print(summary(arraysummary))
        print(countfreq)
        plot(countfreq, xlab = "X (Values of the Random numbers)----------->", ylab =
"FREQUENCY (Of the Random Numbers)----------->", col = rgb(1,0,0,1), main =
paste("HISTOGRAM OF FREQUENCIES (Using Rejection Sampling) : \nNumber of
Random numbers generated = ", num))
        dev.copy(jpeg, "Q3b.jpeg")
        dev.off()
}

generateDISACCREJ(10)

”
```

<u>Mean and Variance</u> :

1) Simulated mean is 3
2) Simulated variance is 2.5
3) Calculated mean is 3
4) Calculated variance is 1
5) Acceptance probability theoretical 0.4444444
6) Acceptance probability simulated 0.5
7) Percentage error in acceptance probability calculated 12.5 %

(This deviation occurs because only 10 numbers are generated)

HISTOGRAM OF FREQUENCIES (Using Rejection Sampling) :
Number of Random numbers generated = 10