Monte Carlo Simulation Lab Assignment-5

Yash Vanjani
(140123046)
Mathematics and Computing
IIT Guwahati

March 9th, 2016

Q 1 Use the Box-Muller method and Marsaglia-Bray method to do the following :
(a) Generate a sample of 100, 500 and 10000 values from N (0, 1). Hence find the sample mean and variance.
(b) Draw histogram in all cases.

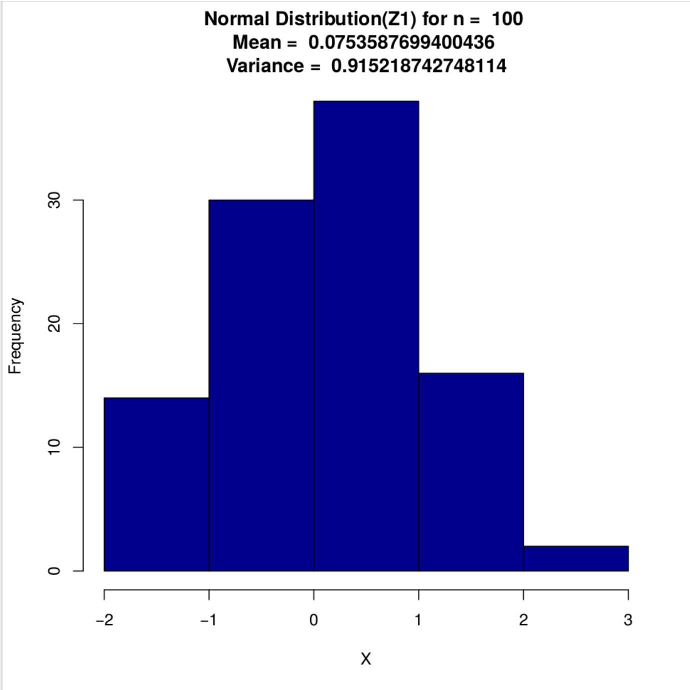
Code of BOX-MULLER for R :

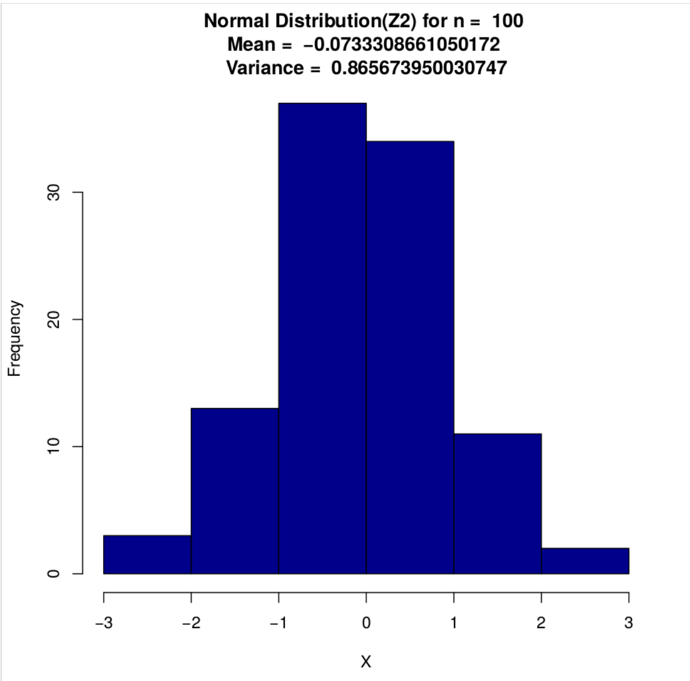
```
f<-function(x1,x2)
{
    return(sqrt(-2*log(x1))*cos(2*pi*x2))
}
g<-function(x1,x2)
{
    return(sqrt(-2*log(x1))*sin(2*pi*x2))
}
p<-c(100,500,10000)
z1<-vector("numeric")
z2<-vector("numeric")
for( i in 1:3)
{
    z1<-vector("numeric")
    z2<-vector("numeric")
    u1<-runif(p[i])
    u2<-runif(p[i])
    j<-1
    while(j<=p[i])
    {
        z1[j]=f(u1[j],u2[j])
        z2[j]=g(u1[j],u2[j])
        j=j+1
    }
    cat("Mean of ",p[i],"numbers of z1 is ",mean(z1),"\n")
    cat("Mean of ",p[i],"numbers of z2 is ",mean(z2),"\n")
    cat("Variance of ",p[i],"numbers of z1 is ",var(z1),"\n")
    cat("Variance of ",p[i],"numbers of z2 is ",var(z2),"\n")
    hist(z1,col="darkblue",main=paste("Normal Distribution(Z1) for n = ",
        p[i],"\nMean = ",mean(z1),"\nVariance = ",var(z1)),xlab="X",ylab=
        "Frequency",breaks=p[i]/20)
    hist(z2,col="darkblue",main=paste("Normal Distribution(Z2) for n = ",
        p[i],"\nMean = ",mean(z2),"\nVariance = ",var(z2)),xlab="X",ylab=
        "Frequency",breaks=p[i]/20)

}
```

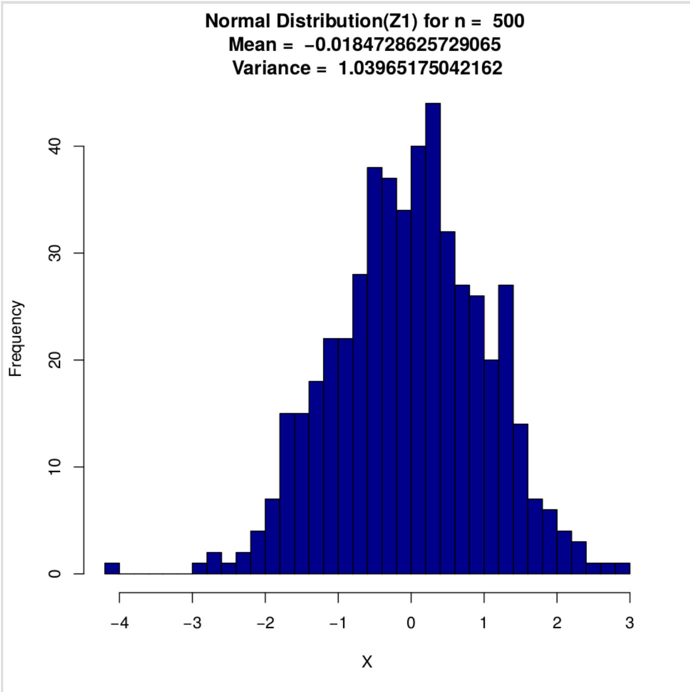
Output :

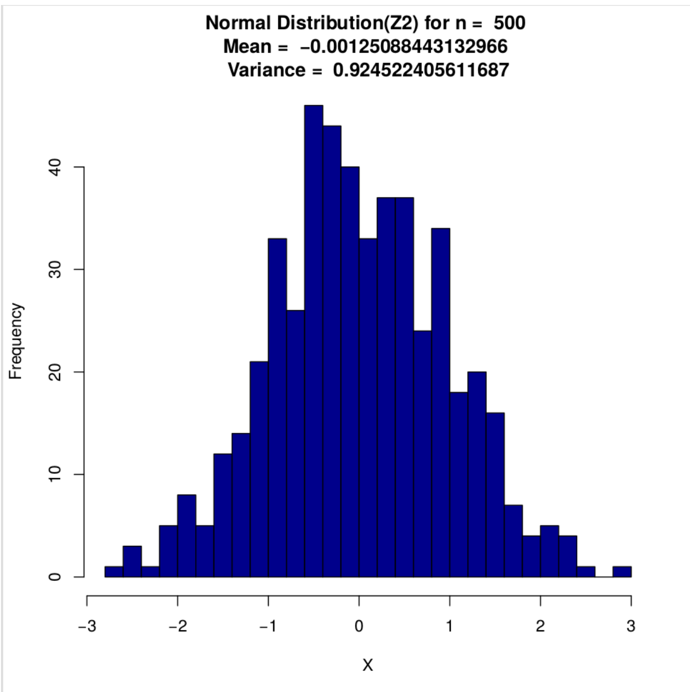
```
Mean of  100 numbers of z1 is  0.075\\
Mean of  100 numbers of z2 is  -0.073 \\
Variance of  100 numbers of z1 is  0.915\\
Variance of  100 numbers of z2 is  0.865 \\\\
Mean of  500 numbers of z1 is  -0.018 \\
Mean of  500 numbers of z2 is  -0.001 \\
Variance of  500 numbers of z1 is  1.039\\
Variance of  500 numbers of z2 is  0.924 \\\\
Mean of  10000 numbers of z1 is  -0.002 \\
Mean of  10000 numbers of z2 is  0.005 \\
Variance of  10000 numbers of z1 is  1.003\\
Variance of  10000 numbers of z2 is  0.999 \\\\
```

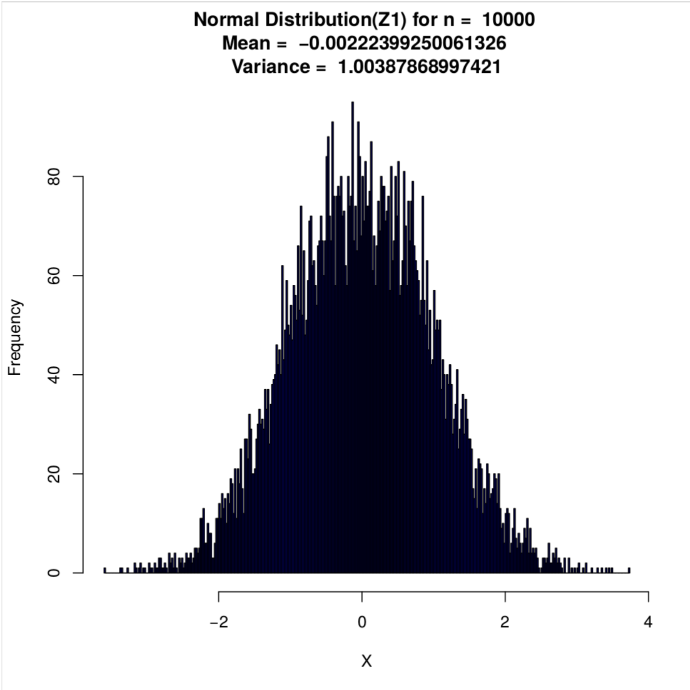
The histograms are shown below:

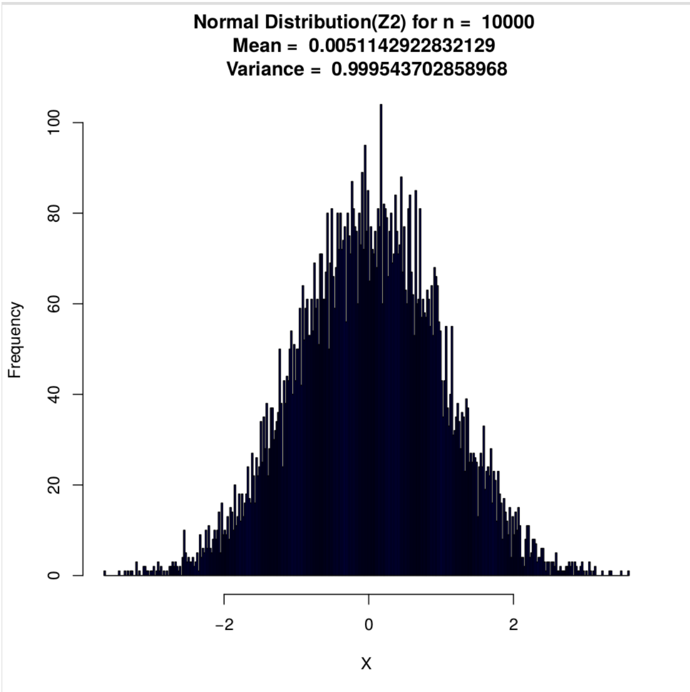





Normal Distribution(Z1) for n = 500
Mean = −0.0184728625729065
Variance = 1.03965175042162
Frequency
0
10
20
30
40
−4
−3
−2
−1
0
1
2
3
X


Normal Distribution(Z2) for n = 500
Mean = −0.00125088443132966
Variance = 0.924522405611687
Frequency
0
10
20
30
40
−3
−2
−1
0
1
2
3
X


Normal Distribution(Z1) for n = 10000
Mean = −0.00222399250061326
Variance = 1.00387868997421
Frequency
0
20
40
60
80
−2
0
2
4
X


Normal Distribution(Z2) for n = 10000
Mean = 0.0051142922832129
Variance = 0.999543702858968
Frequency
0
20
40
60
80
100
−2
0
2
X

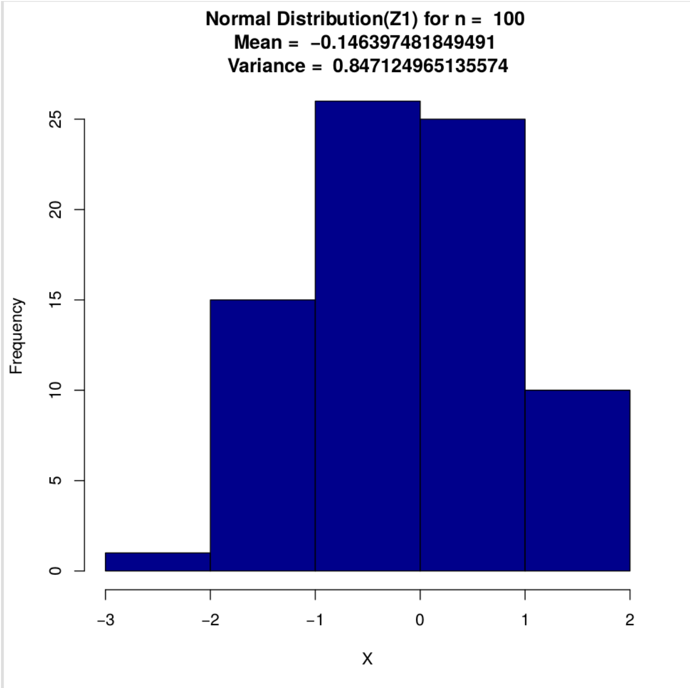
The code for MARSAGLIA-BRAY for R:

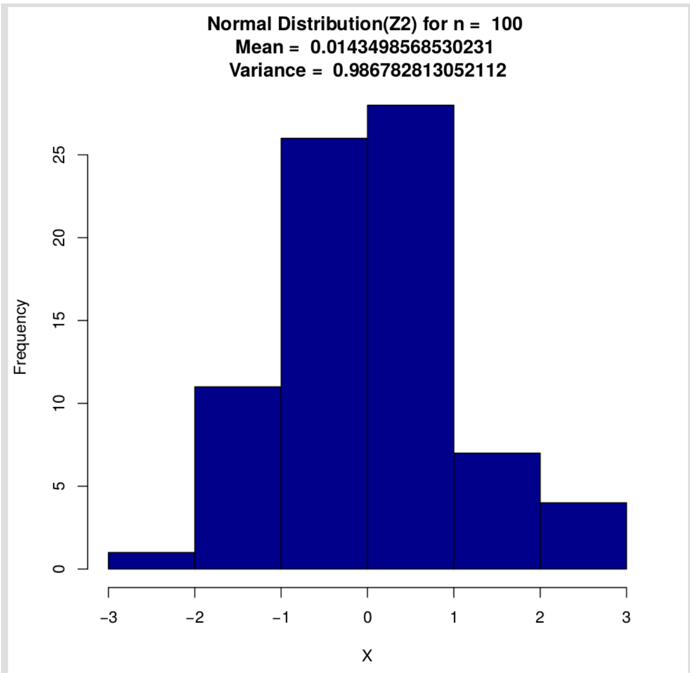
```
f<-function(v1,v2)
{
    return(v1*sqrt(-2*log(v1^2+v2^2)/(v1^2+v2^2)))
}
g<-function(v1,v2)
{
    return(v2*sqrt(-2*log(v1^2+v2^2)/(v1^2+v2^2)))
}
p<-c(100,500,10000)
for(i in 1:3)
{
    v1<-2*runif(p[i])-1
    v2<-2*runif(p[i])-1
    j<-1
    k<-1
    z1<-vector("numeric")
    z2<-vector("numeric")
    while(j<=p[i])
    {
        if(v1[j]^2+v2[j]^2<1)
        {
            z1[k]<-f(v1[j],v2[j])
            z2[k]<-g(v1[j],v2[j])
            k=k+1
        }
        j=j+1
    }
    cat("Mean of ",p[i],"numbers of z1 is ",mean(z1),"\n")
    cat("Mean of ",p[i],"numbers of z2 is ",mean(z2),"\n")
    cat("Variance of ",p[i],"numbers of z1 is ",var(z1),"\n")
    cat("Variance of ",p[i],"numbers of z2 is ",var(z2),"\n")
    hist(z1,col="darkblue",main=paste("Normal Distribution(Z1) for n = ",
        p[i],"\nMean = ",mean(z1),"\nVariance = ",var(z1)),xlab="X",ylab=
        "Frequency",breaks=p[i]/20)
    hist(z2,col="darkblue",main=paste("Normal Distribution(Z2) for n = ",
        p[i],"\nMean = ",mean(z2),"\nVariance = ",var(z2)),xlab="X",ylab=
        "Frequency",breaks=p[i]/20)
}
```

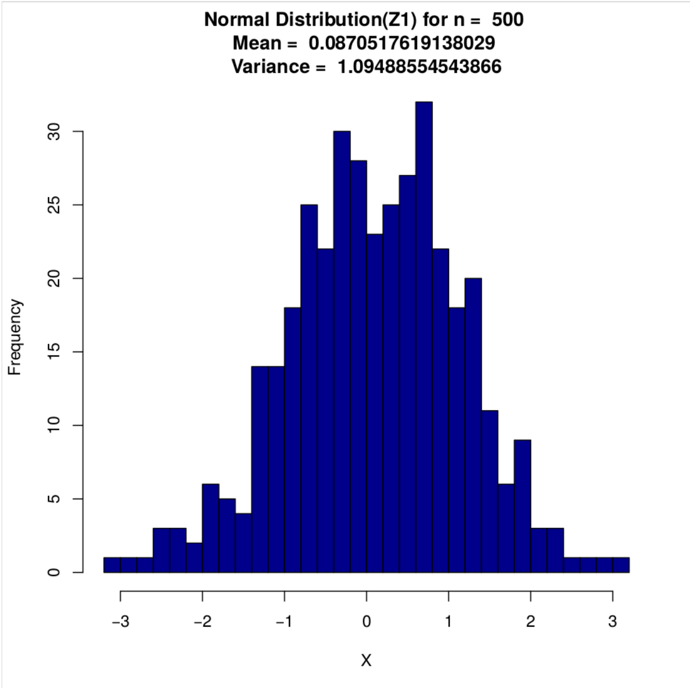
Output :

```
Mean of 100 numbers of z1 is −0.146\\
Mean of 100 numbers of z2 is 0.014 \\
Variance of 100 numbers of z1 is 0.847\\
Variance of 100 numbers of z2 is 0.986 \\\\
Mean of 500 numbers of z1 is 0.087 \\
Mean of 500 numbers of z2 is 0.012 \\
Variance of 500 numbers of z1 is 1.094\\
Variance of 500 numbers of z2 is 0.054 \\\\
Mean of 10000 numbers of z1 is 0.012 \\
Mean of 10000 numbers of z2 is 0.004 \\
Variance of 10000 numbers of z1 is 0.984\\
Variance of 10000 numbers of z2 is 1.040\\\\
```

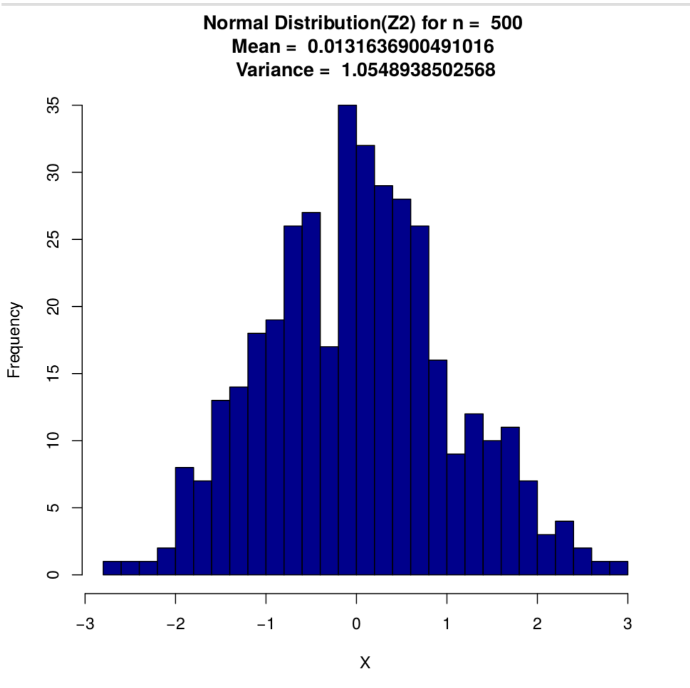
The histograms are shown below:



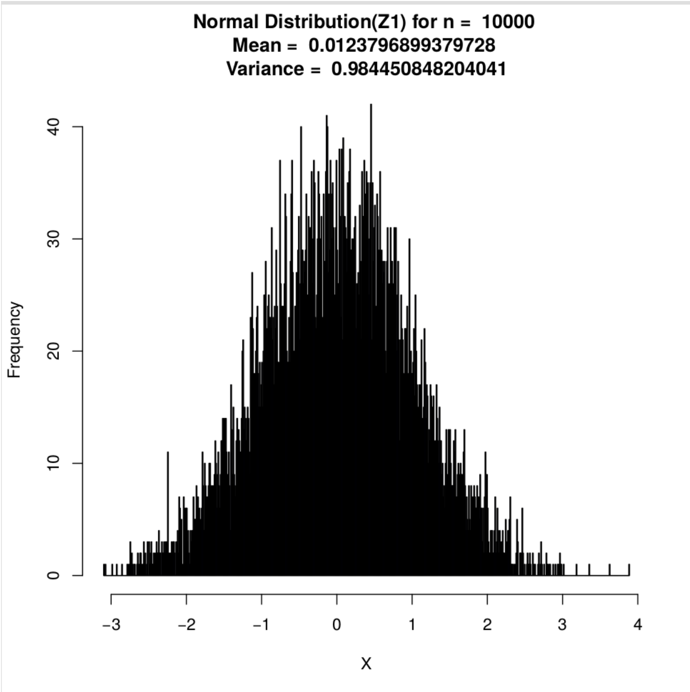


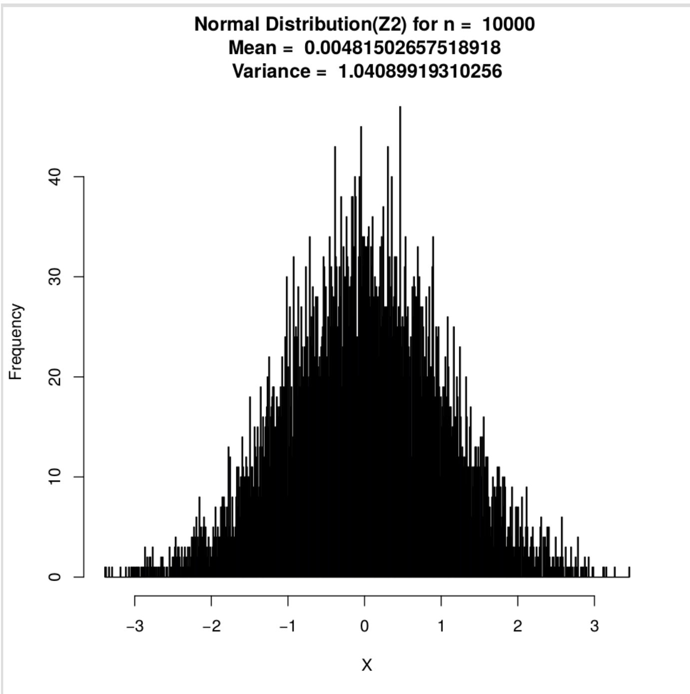


Normal Distribution(Z1) for n = 500
Mean = 0.0870517619138029
Variance = 1.09488554543866
Frequency
0
5
10
15
20
25
30
−3
−2
−1
0
1
2
3
X



Normal Distribution(Z2) for n = 500
Mean = 0.0131636900491016
Variance = 1.0548938502568
Frequency
0
5
10
15
20
25
30
35
−3
−2
−1
0
1
2
3
X


Normal Distribution(Z1) for n = 10000
Mean = 0.0123796899379728
Variance = 0.984450848204041
Frequency
0
10
20
30
40
−3
−2
−1
0
1
2
3
4
X


Normal Distribution(Z2) for n = 10000
Mean = 0.00481502657518918
Variance = 1.04089919310256
Frequency
0
10
20
30
40
−3
−2
−1
0
1
2
3
X

Observations :

1. The mean of the distribution calculated using both the methods comes out to be nearly equal to 0 which is in fact the mean of Standard Normal Distribution.

2. The variance of these values indicate that they are going towards 1 which is equal to the theoretical value of the variance and thus the given Random variable actually has a distribution which is the same as the Normal Distribution with variance 1 and mean 0. Furthermore the histograms resemble those of Normal Distribution.

3. There are two methods which can be used to generate the Normal Distribution: Box-Muller and Marsaglia-Bray, both of them as shown above give nearly same precision but in the first one all the random numbers generated are used while in the other method the random numbers generated are checked whether that they are inside the circle of radius 1 before proceding further.

4. When the number of sample random variables is taken to as large as 10,000, the normal distribution of Random variables becomes visible.

5. The probability density graph is symmetric about mean.

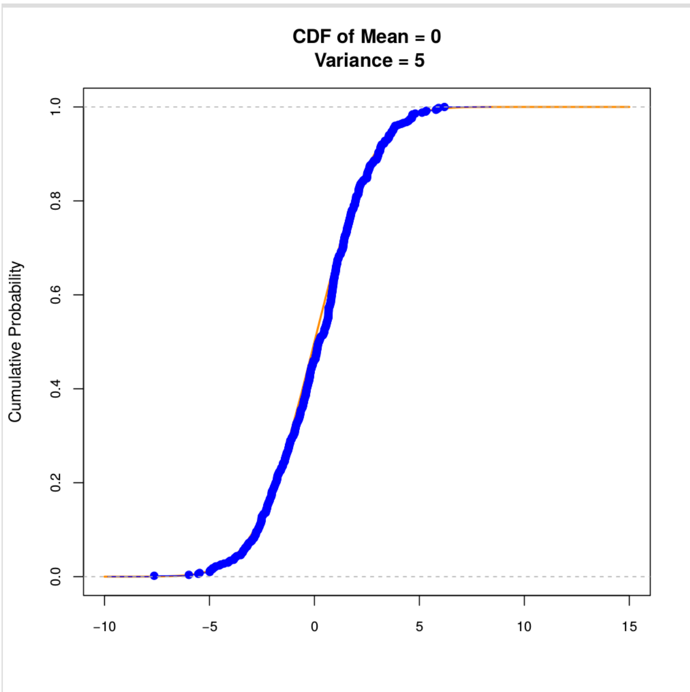
Q 2 Now use the above generated values to generated samples from $N(\mu = 0; \sigma^2 = 5)$ and $N(\mu = 5; \sigma^2 = 5)$. Hence plot the empirical(from sample with size 500) distribution function and theoretical distribution function in the same plot.

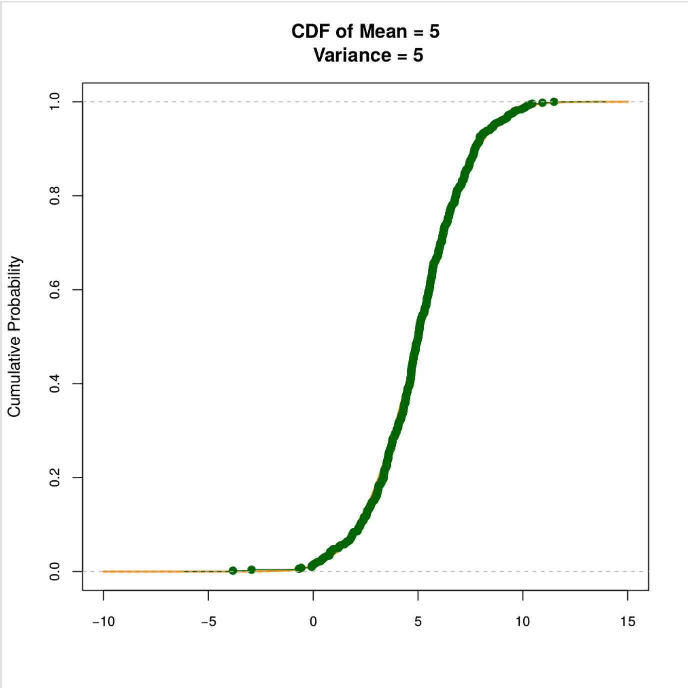
Code for R:

```
f<-function(x1,x2)
{
    return(sqrt(-2*log(x1))*cos(2*pi*x2))
}
g<-function(x1,x2)
{
    return(sqrt(-2*log(x1))*sin(2*pi*x2))
}
p<-c(100,500,1000)
z1<-vector("numeric")
z2<-vector("numeric")
for( i in 1:3)
{
    z1<-vector("numeric")
    z2<-vector("numeric")
    u1<-runif(p[i])
    u2<-runif(p[i])
    j<-1
    while(j<=p[i])
    {
        z1[j]=sqrt(5)*f(u1[j],u2[j])
        z2[j]=(sqrt(5)*g(u1[j],u2[j]))+5
        j=j+1
    }
    cat("Mean of ",p[i],"numbers of z1 is ",mean(z1),"\n")
    cat("Mean of ",p[i],"numbers of z2 is ",mean(z2),"\n")
    cat("Variance of ",p[i],"numbers of z1 is ",var(z1),"\n")
    cat("Variance of ",p[i],"numbers of z2 is ",var(z2),"\n")
    #hist(z1,col="darkblue",main=paste("Normal Distribution(Z1) for n = "
        ,p[i],"\nMean = ",mean(z1),"\nVariance = ",var(z1)),xlab="X",ylab
        ="Frequency",breaks=p[i]/20)
    #hist(z2,col="darkblue",main=paste("Normal Distribution(Z2) for n = "
        ,p[i],"\nMean = ",mean(z2),"\nVariance = ",var(z2)),xlab="X",ylab
        ="Frequency",breaks=p[i]/20)

    if(p[i]==500)
    {
        xseq<-seq(-10,15,0.01);
        h=ecdf(z1)
        m=ecdf(z2)
          cumulative<-pnorm(xseq,0,sqrt(5))
```

```
        plot(xseq, cumulative, col="darkorange", xlab="", ylab="
            Cumulative Probability",type="l",lwd=2, cex=2, main="CDF of
            Mean = 0\n Variance = 5", cex.axis=.8)
    lines(h,col="blue")

    cum<-pnorm(xseq,5,sqrt(5))
    plot(xseq, cum, col="darkorange", xlab="", ylab="Cumulative
        Probability",type="l",lwd=2, cex=2, main="CDF of Mean = 5\n
        Variance = 5", cex.axis=.8)
    lines(m,col="darkgreen")
    }
}
```

Plots of the emperical distribution function and theoritical distribution function is as follows:





CDF of Mean = 5
Variance = 5
Cumulative Probability
0.0
0.2
0.4
0.6
0.8
1.0
−10
−5
0
5
10
15

Q 3 Keep a track of the computational time required for both the methods. Which method is faster ?

Code of R for BOX-MULLER method :

```
f<-function(x1,x2)
{
    return(sqrt(-2*log(x1))*cos(2*pi*x2))
}
g<-function(x1,x2)
{
    return(sqrt(-2*log(x1))*sin(2*pi*x2))
}
p<-c(10^3,10^4,10^5)
z1<-vector("numeric")
z2<-vector("numeric")
for( i in 1:3)
{
    ptm<-proc.time()
    z1<-vector("numeric")
    z2<-vector("numeric")
    u1<-runif(p[i])
    u2<-runif(p[i])
    j<-1
    while(j<=p[i])
    {
        z1[j]=f(u1[j],u2[j])
        z2[j]=g(u1[j],u2[j])
        j=j+1
    }
    cat("\n For n = ",p[i],"The time taken by Box muller method\n")
    print(proc.time() - ptm)

}
```

<u>Output :</u>

	USER	SYSTEM	ELAPSED
n=100	0.0110	0.000	0.011
n=500	0.254	0.000	0.253
n=10000	31.498	2.757	34.241

Table 1: Time taken by BOX-MULLER method

Code of R for MARSAGLIA-GRAY method :

```
f<-function(v1,v2)
{
    return(v1*sqrt(-2*log(v1^2+v2^2)/(v1^2+v2^2)))
}
g<-function(v1,v2)
{
    return(v2*sqrt(-2*log(v1^2+v2^2)/(v1^2+v2^2)))
}
p<-c(100,500,10000)
for(i in 1:3)
{
    ptm<-proc.time()
    v1<-2*runif(p[i])-1
    v2<-2*runif(p[i])-1
    j<-1
    k<-1
    z1<-vector("numeric")
    z2<-vector("numeric")
    while(j<=p[i])
    {
        if(v1[j]^2+v2[j]^2<1)
        {
            z1[k]<-f(v1[j],v2[j])
            z2[k]<-g(v1[j],v2[j])
            k=k+1
        }
        j=j+1
    }
    cat("\n For n = ",p[i]," The time taken by Marsaglia Bray method\n")
    print(proc.time() - ptm)
}
```

Output :

	USER	SYSTEM	ELAPSED
n=100	0.001	0.000	0.001
n=500	0.005	0.001	0.006
n=10000	0.198	0.007	0.206

Table 2: Time taken by MARSAGLIA-BRAY method method

Observation :

1. As expected we observe that MARSAGLIA-BRAY method takes very very less time than BOX-MULLER method to compute random numbers.
2. MARSAGLIA-BRAY method is almost 150x faster than BOX-MULLER method.

Q 4 For the Marsaglia-Bray method keep track of the proportional of values rejected. How does it compare with $1 - \frac{\pi}{4}$?

Code of R for MARSAGLIA-GRAY method :

```
f<-function(v1,v2)
{
    return(v1*sqrt(-2*log(v1^2+v2^2)/(v1^2+v2^2)))
}
g<-function(v1,v2)
{
    return(v2*sqrt(-2*log(v1^2+v2^2)/(v1^2+v2^2)))
}
p<-c(100,500,10000)
for(i in 1:3)
{
    accepted<-array(0,p[i])
    v1<-2*runif(p[i])-1
    v2<-2*runif(p[i])-1
    j<-1
    k<-1
    z1<-vector("numeric")
    z2<-vector("numeric")
    while(j<=p[i])
    {
        if(v1[j]^2+v2[j]^2<1)
        {
            z1[k]<-f(v1[j],v2[j])
            z2[k]<-g(v1[j],v2[j])
            k=k+1
            accepted[j]=accepted[j]+1

        }
        j=j+1
    }
    cat("Number of values accepted for ",p[i],"numbers is ",sum(accepted),"\n")
    cat("Rejection probability =  ",1-(sum(accepted)/p[i]),"\n")

}
```

Output :

	Rejection Probability
n=100	0.16
n=500	0.216
n=10000	0.2187

Table 3: Rejection Probabaility