MA226 : Monte-Carlo Simulation
Generating Standard Normal Distritution
Assignment 6

Turkhade Hrushikesh Pramod
150123044

09-03-2017

1 Problem 1

We have to generate sample from standard normal distribution using Box-Muller Transform and Marsaglia-Bray Method.

1.1 Source code of the solution

```
n<-100

string<-"hund"

#Box-Muller
u1<-runif(n)
u2<-runif(n)

z1<-sqrt(-2*log(u1))*cos(2*pi*u2)
z2<-sqrt(-2*log(u1))*sin(2*pi*u2)

sample1<-c(z1,z2)

png(paste("que1_",string,"1.png"))
hist(sample1,breaks=50,col="cyan",plot=TRUE)


#Masaglia-Bray
i<-1

r1<-vector(length=n)
r2<-vector(length=n)

while(i<=n){
        u1<-runif(1)
        u2<-runif(1)

        u1<-2*u1-1
        u2<-2*u2-1

        x<-u1*u1+u2*u2
        if(x < 1){
                temp<-sqrt(-2*log(x)/x)
                r1[i]<-u1*temp
                r2[i]<-u2*temp
                i<-i+1
        }
}

sample2<-c(r1,r2)

png(paste("que1_",string,"2.png"))
hist(sample2,breaks=50,col="cyan",plot=TRUE)
```

1.2 Generated Histograms

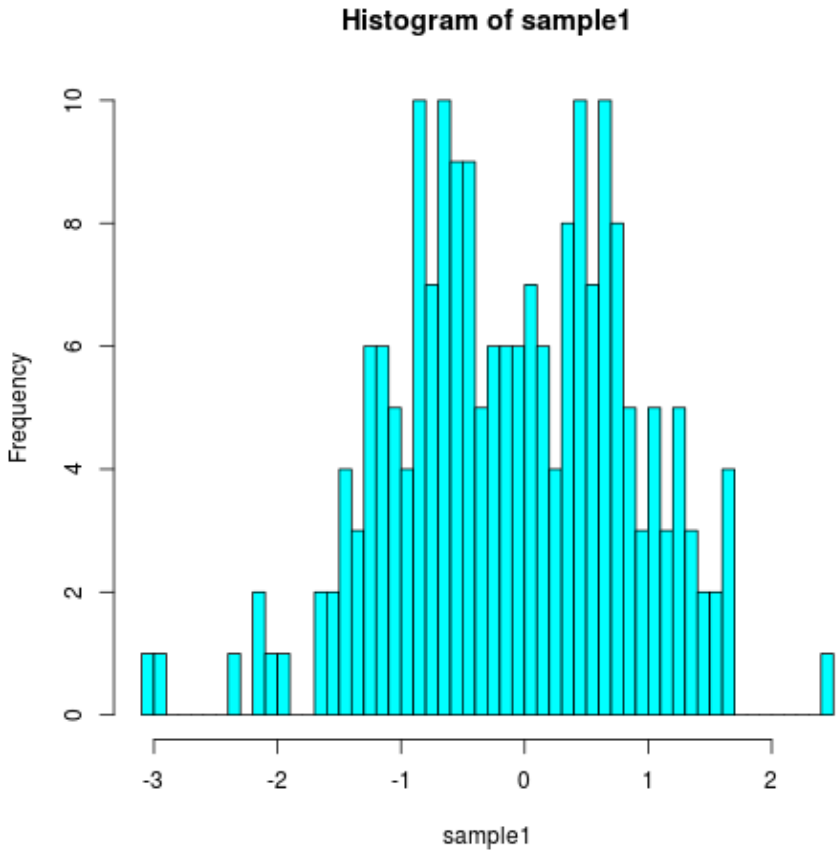

Abbildung 1: Using Box-Muller for 100 sample

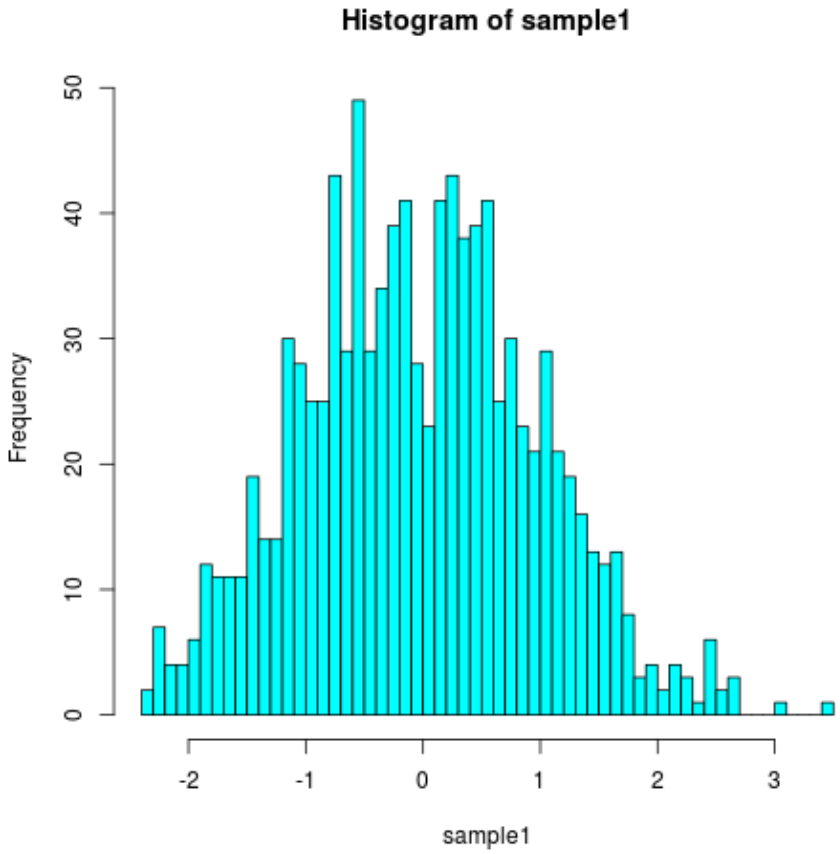

Abbildung 2: Using Box-Muller for 500 sample

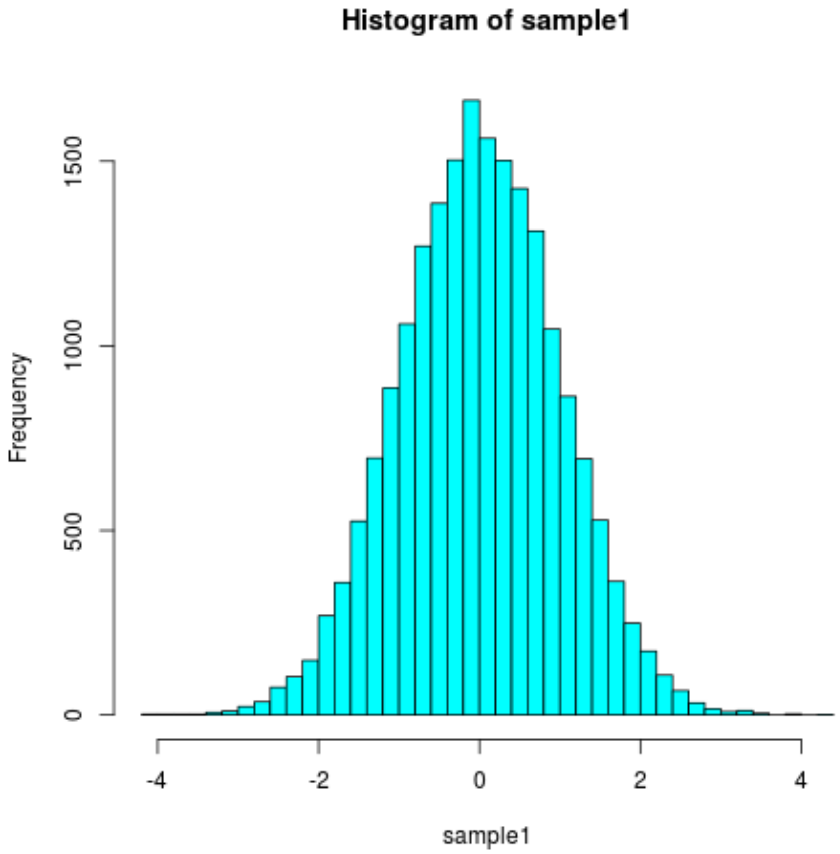

Abbildung 3: Using Box-Muller for 10000 sample

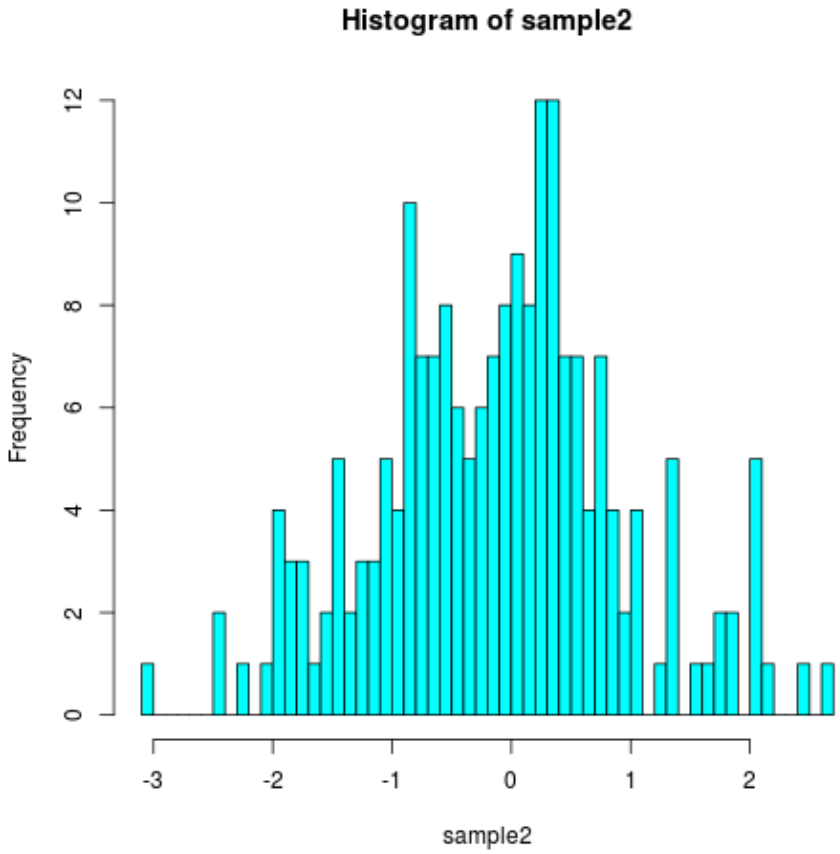

Abbildung 4: Using Marsaglia-Bray for 100 sample

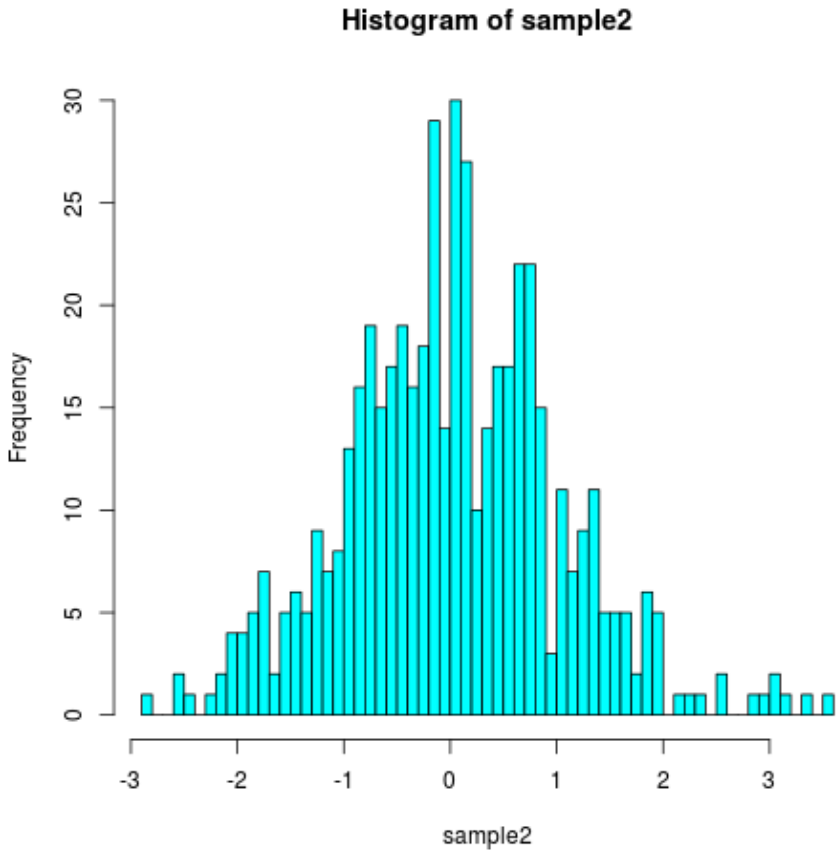

Abbildung 5: Using Marsaglia-Bray for 500 sample

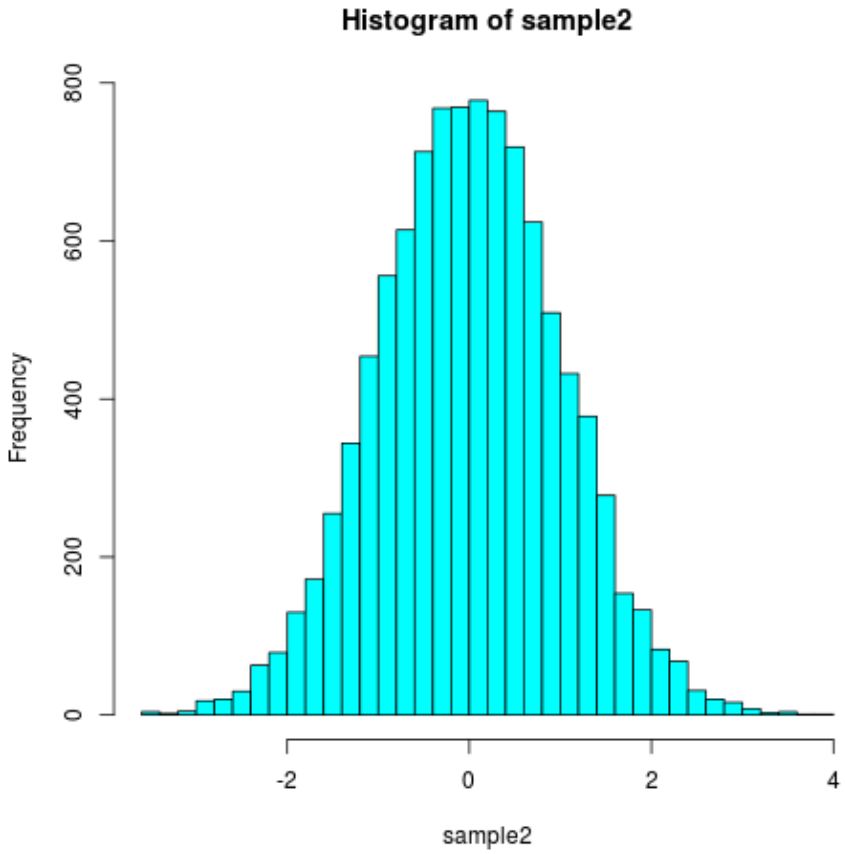

Abbildung 6: Using Marsaglia-Bray for 10000 sample

2 Problem 2

In this problem we have to plot emirical and theoritical distribution of $N(0,5)$ and $N(5,5)$.

2.1 Source code generating Empirical and Theoritical Distributions

```
n<-100

string<-"ten-thou"


#Box-Muller
u1<-runif(n)
u2<-runif(n)

z1<-sqrt(-2*log(u1))*cos(2*pi*u2)
z2<-sqrt(-2*log(u1))*sin(2*pi*u2)

sample1<-c(z1,z2)

samplex<-sqrt(5)*sample1
sampley<-sqrt(5)*sample1+5


png("que2_one.png")
plot(ecdf(samplex),col="red")
par(new=TRUE)
samplex<-sort(samplex)
plot(pnorm(samplex,mean=0,sd=sqrt(5)),col="cyan")


png("que2_two.png")
plot(ecdf(sampley),col="red")
par(new=TRUE)
sampley<-sort(sampley)
plot(pnorm(sampley,mean=5,sd=sqrt(5)),col="cyan")
```

2.2 Generated Graphs

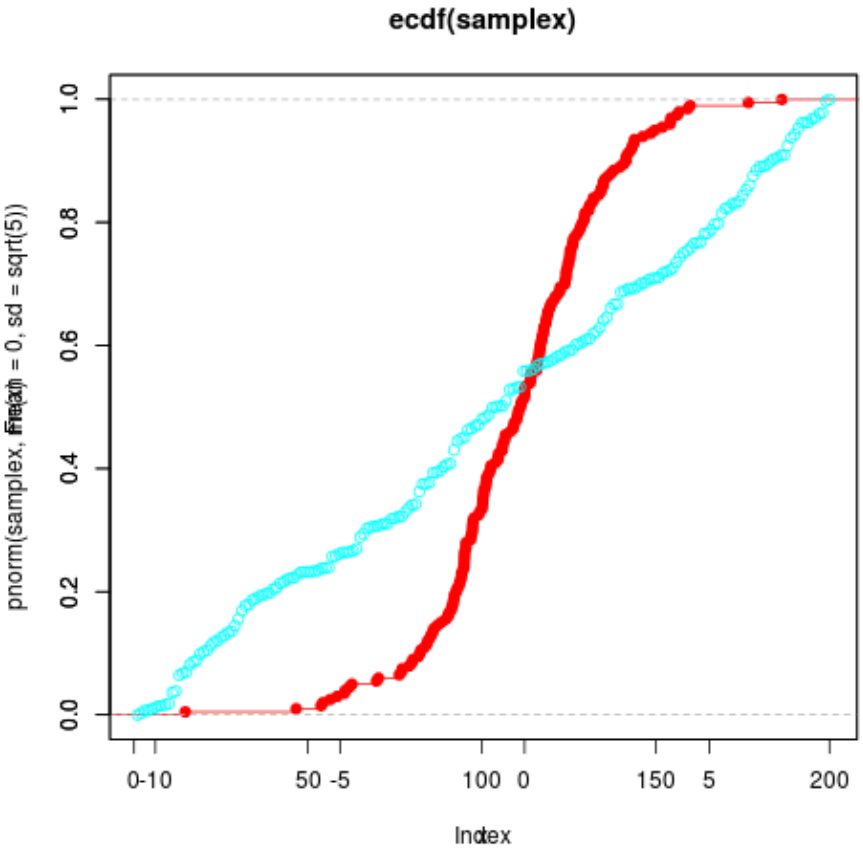

Abbildung 7: Plot of Emirical and Theoritical distribution of $N(0, 5)$

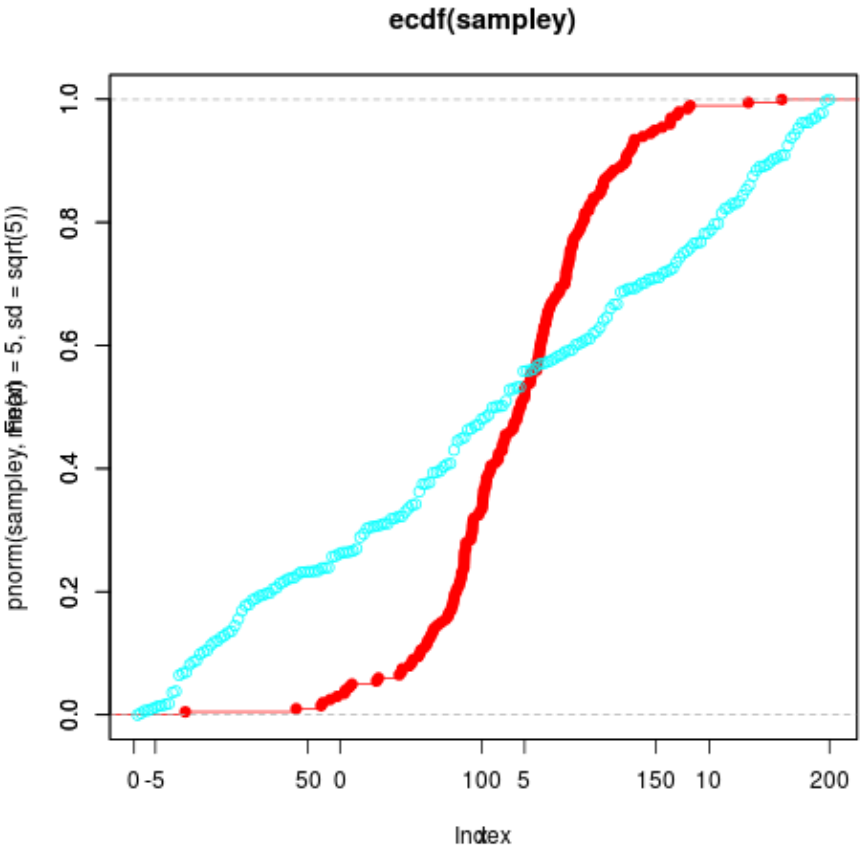

Abbildung 8: Plot of Emirical and Theoritical distribution of $N(5, 5)$

3 Problem 3

In this problem we have to calculate the time taken to calculate Standard Normal distribution using Box-Muller and Marsaglia-Bray Mehods.

3.1 Source code for Box-Muller

```
radius<-function(u){
        return (-2*log(u))
}
arg<-function(u){
        return (2*pi*u)
}
n<-10000
stime<-Sys.time();
sample1<-c()
sample2<-c()
for(i in 1:n/2){
        u1<-runif(1)
        u2<-runif(1)
        sample1[i]<-sqrt(radius(u1))*cos(arg(u2))
        sample2[i]<-sqrt(radius(u1))*sin(arg(u2))
}


sample<-c(sample1,sample2)
etime<-Sys.time();
cat("Computation Time of Box-Muller Method: ",etime-stime,"\n");
```

3.2 Source code for Marsaglia-Bray

```
square<-function(u1,u2){
        return (u1**2+u2**2)
}
temp<-function(x){
        return (sqrt(-2*log(x)/x))
}

n<-10000
stime<-Sys.time();
sample1<-c()
sample2<-c()
i<-1
while(n>=0){
        u1<-runif(1)
        u2<-runif(1)
        u1<-2*u1-1
        u2<-2*u2-1
        x<-square(u1,u2)
        if(x>1){
                next
        }
        y<-temp(x)
        sample1[i]<-u1*y
        sample2[i]<-u2*y
```

```
        i<-i+1
        n<-n-2
}
sample<-c(sample1,sample2)
etime<-Sys.time();
cat("Computation Time of Marsaglia-Bray Method: ",etime-stime,"\n");
```

3.3 Analysis

Time taken for Box-Muller: 0.0003612041

Time taken for Masaglia-Bray: 0.002064705

We can cleary see that Box-Muller takes more time than Marsaglia Bray

4 Problem 4

In this section we calculate number of rejections proportional to total sampled uniform random variables.

4.1 Source code for Solution

```
square<-function(u1,u2){
        return (u1**2+u2**2)
}
temp<-function(x){
        return (sqrt(-2*log(x)/x))
}

rejected<-0
n<-10000
ct<-n
stime<-Sys.time();
sample1<-c()
sample2<-c()
i<-1
while(n>=0){
        u1<-runif(1)
        u2<-runif(1)
        u1<-2*u1-1
        u2<-2*u2-1
        x<-square(u1,u2)
        if(x>1){
                rejected<-rejected+1
                next
        }
        y<-temp(x)
        sample1[i]<-u1*y
        sample2[i]<-u2*y
        i<-i+1
        n<-n-2
}
```

```
cat("Rejection Percentage: ",rejected/(rejected+(ct/2)),"\n")
```

4.2 Analysis

Here, rejection probability comes as 0.2092361 which is quite close to $1 - \frac{pi}{4} = 0.2146$