Assignment-5

Abheek Ghosh
140123047

March 8, 2016

1 Box-Muller Code

Code for R

```r
start.time <- Sys.time()
#Box Muller method
n <- 10000 #no of values
u1 <- runif(n)
u2 <- runif(n)

temp <- ((-2) * log(u1))^(1/2)

z1 <- temp * cos(2 * pi * u2)
z2 <- temp * sin(2 * pi * u2)

end.time <- Sys.time()
time.taken <- end.time - start.time


cat("Box-Muller Method\n")
cat("Time taken for 10000 numbers generated = ", time.taken,"\n")
cat("For 100 values\n")
cat("Sample's Mean, X = ", mean(z1[1:100]), ", Y = ", mean(z2[1:100]), "\n")
cat("Sample's Variance, X = ", var(z1[1:100]), ", Y = ", var(z2[1:100]), "\n")
cat("Sample's Covariance = ", cov(z1[1:100], z2[1:100]), "\n\n")

cat("For 500 values\n")
cat("Sample's Mean, X = ", mean(z1[1:500]), ", Y = ", mean(z2[1:500]), "\n")
cat("Sample's Variance, X = ", var(z1[1:500]), ", Y = ", var(z2[1:500]), "\n")
cat("Sample's Covariance = ", cov(z1[1:500], z2[1:500]), "\n\n")

cat("For 10000 values\n")
cat("Sample's Mean, X = ", mean(z1), ", Y = ", mean(z2), "\n")
cat("Sample's Variance, X = ", var(z1), ", Y = ", var(z2), "\n")
cat(”Sample’s Covariance = ”, cov(z1, z2), ”\n\n”)

plot(z1[1:100], z2[1:100], col=’black’,  main=”2–D Standard Normal Distribution, 100 values,”, xlab=”X”, ylab=”Y”)
dev.copy(png,”plot1_1.png”);
dev.off ();

hist(z1[1:100], main=”Standard Normal Distribution, X, 100 values”, xlab=”Range of random numbers”, ylab=”Density”, breaks=15)
dev.copy(png,”plot1_1_X.png”);
dev.off ();

hist(z2[1:100], main=”Standard Normal Distribution, Y, 100 values”, xlab=”Range of random numbers”, ylab=”Density”, breaks=15)
dev.copy(png,”plot1_1_Y.png”);
dev.off ();



plot(z1[1:500], z2[1:500], col=’black’, cex=0.5, main=”2–D Standard Normal Distribution, 500 values”, xlab=”X”, ylab=”Y”)
dev.copy(png,”plot1_2.png”);
dev.off ();

hist(z1[1:500], main=”Standard Normal Distribution, X, 500 values”, xlab=”Range of random numbers”, ylab=”Density”, breaks=20)
dev.copy(png,”plot1_2_X.png”);
dev.off ();

hist(z2[1:500], main=”Standard Normal Distribution, Y, 500 values”, xlab=”Range of random numbers”, ylab=”Density”, breaks=20)
dev.copy(png,”plot1_2_Y.png”);
dev.off ();



plot(z1, z2, pch=16, col=’black’, cex=0.5, main=”2–D Standard Normal Distribution, 10000 values”, xlab=”X”, ylab=”Y”)
dev.copy(png,”plot1_3.png”);
dev.off ();

hist(z1, main=”Standard Normal Distribution, X, 10000 values”, xlab=”Range of random numbers”, ylab=”Density”, breaks=25)
dev.copy(png,”plot1_3_X.png”);
dev.off ();

hist(z2, main=”Standard Normal Distribution, Y, 10000 values”, xlab=”Range of random numbers”, ylab=”Density”, breaks=25)
```

```
dev.copy(png,”plot1_3_Y.png”);
dev.off ();


norm <- function(x, mu, sig) {
    return ( (1/ ((2*pi)^(1/2) * sig )) * exp ( -((x-mu)/sig)^2 / 2) )
}


##For N(  = 0,   =5)
x1 <- z1[1:500] * 5^(1/2)
y1 <- z2[1:500] * 5^(1/2)

cat(”N(  = 0,   =5) for 500 values\n”)
cat(”Sample’s Mean, X = ”, mean(x1[1:500]), ”, Y = ”, mean(y1[1:500]), ”\n”)
cat(”Sample’s Variance, X = ”, var(x1[1:500]), ”, Y = ”, var(y1[1:500]), ”\n”)
cat(”Sample’s Covariance = ”, cov(x1[1:500], y1[1:500]), ”\n\n”)

plot(x1[1:500], y1[1:500], col=’black’, cex=0.5, main=”2-D N(  = 0,   =5), 500 values”, xlab=”X”, ylab=”Y”)
dev.copy(png,”plot3_1.png”);
dev.off ();

hist(x1[1:500], main=”N(  = 0,   =5), X, 500 values”, xlab=”Range of random numbers”, ylab=”Density”, breaks=20)
points(x1 , 500 * norm(x1, 0, 5) , col=’black’, cex=0.5)
dev.copy(png,”plot3_1_X.png”);
dev.off ();

hist(y1[1:500], main=”N(  = 0,   =5), Y, 500 values”, xlab=”Range of random numbers”, ylab=”Density”, breaks=20)
points(y1 , 500 * norm(y1, 0, 5) , col=’black’, cex=0.5)
dev.copy(png,”plot3_1_Y.png”);
dev.off ();


##For N(  = 5,   =5)



x2 <- x1 + 5
y2 <- y1 + 5

cat(”N(  = 5,   =5) for 500 values\n”)
cat(”Sample’s Mean, X = ”, mean(x2[1:500]), ”, Y = ”, mean(y2[1:500]), ”\n”)
cat(”Sample’s Variance, X = ”, var(x2[1:500]), ”, Y = ”, var(y2[1:500]), ”\n”)
cat(”Sample’s Covariance = ”, cov(x2[1:500], y2[1:500]), ”\n\n”)

```

```
plot(x2[1:500], y2[1:500], col='black', cex=0.5, main="2-D N(  = 5,   =5), 500 values", xlab="X", ylab="Y")
dev.copy(png,"plot3_2.png");
dev.off ();

hist(x2[1:500], main="N(  = 5,   =5), X, 500 values", xlab="Range of random numbers", ylab="Density", breaks=20)
points(x2 , 500 * norm(x2, 5, 5) , col='black', cex=0.5)
dev.copy(png,"plot3_2_X.png");
dev.off ();

hist(y2[1:500], main="N(  = 5,   =5), Y, 500 values", xlab="Range of random numbers", ylab="Density", breaks=20)
points(y2 , 500 * norm(y2, 5, 5) , col='black', cex=0.5)
dev.copy(png,"plot3_2_Y.png");
dev.off ();
rm(list = ls())
```

2 Marsaglia Bray Code

Code for R

```
#Marsaglia Bray method
start.time <- Sys.time()
n <- 11000  #no of values
p <- 0
q <- 0
u1 <- vector(,0)
u2 <- vector(,0)

while (q < 10000) {
   p <- p + n
   u <- runif(n)
   v <- runif(n)
   u <- 2*u - 1
   v <- 2*v - 1
   temp <- u^2 + v^2
   u1 <- c( u1, u[temp < 1] )
   u2 <- c( u2, v[temp < 1] )
   q <- length(u1)
   n <- 10000 - q + 1
}

temp1 <- u1[1:1000]^2 + u2[1:1000]^2
temp2 <- ((-2) * log(temp1))^(1/2)
temp1 <- temp1^(1/2)

```

```
z1 <- temp2 * (u1[1:1000] / temp1)
z2 <- temp2 * (u2[1:1000] / temp1)

end.time <- Sys.time()
time.taken <- end.time - start.time
time.taken


cat(”Marsaglia-Bray Method\n”)
cat(”Time taken for 10000 numbers generated = ”, time.taken,”\n”)
cat(”Rejection ratio, Theoretical = ”, 1 - pi/4, ”, Generated = ”, 1 - q/p, ”\n\n”)
cat(”For 100 values\n”)
cat(”Sample’s Mean, X = ”, mean(z1[1:100]), ”, Y = ”, mean(z2[1:100]), ”\n”)
cat(”Sample’s Variance, X = ”, var(z1[1:100]), ”, Y = ”, var(z2[1:100]), ”\n”)
cat(”Sample’s Covariance = ”, cov(z1[1:100], z2[1:100]), ”\n\n”)

cat(”For 500 values\n”)
cat(”Sample’s Mean, X = ”, mean(z1[1:500]), ”, Y = ”, mean(z2[1:500]), ”\n”)
cat(”Sample’s Variance, X = ”, var(z1[1:500]), ”, Y = ”, var(z2[1:500]), ”\n”)
cat(”Sample’s Covariance = ”, cov(z1[1:500], z2[1:500]), ”\n\n”)

cat(”For 10000 values\n”)
cat(”Sample’s Mean, X = ”, mean(z1), ”, Y = ”, mean(z2), ”\n”)
cat(”Sample’s Variance, X = ”, var(z1), ”, Y = ”, var(z2), ”\n”)
cat(”Sample’s Covariance = ”, cov(z1, z2), ”\n\n”)

plot(z1[1:100], z2[1:100], col=’black’,  main=”2-D Standard Normal Distribution, 100 values,”, xlab=”X”, ylab=”Y”)
dev.copy(png,”plot2_1.png”);
dev.off ();

hist(z1[1:100], main=”Standard Normal Distribution, X, 100 values”, xlab=”Range of random numbers”, ylab=”Density”, breaks=15)
dev.copy(png,”plot2_1_X.png”);
dev.off ();

hist(z2[1:100], main=”Standard Normal Distribution, Y, 100 values”, xlab=”Range of random numbers”, ylab=”Density”, breaks=15)
dev.copy(png,”plot2_1_Y.png”);
dev.off ();



plot(z1[1:500], z2[1:500], col=’black’, cex=0.5, main=”2-D Standard Normal Distribution, 500 values”, xlab=”X”, ylab=”Y”)
dev.copy(png,”plot2_2.png”);
dev.off ();

```

```
hist(z1[1:500], main=”Standard Normal Distribution, X, 500 values”, xlab=”Range of random numbers”, ylab=”Density”, breaks=20)
dev.copy(png,”plot2_2_X.png”);
dev.off ();

hist(z2[1:500], main=”Standard Normal Distribution, Y, 500 values”, xlab=”Range of random numbers”, ylab=”Density”, breaks=20)
dev.copy(png,”plot2_2_Y.png”);
dev.off ();



plot(z1, z2, pch=16, col='black', cex=0.5, main=”2-D Standard Normal Distribution, 10000 values”, xlab=”X”, ylab=”Y”)
dev.copy(png,”plot2_3.png”);
dev.off ();

hist(z1, main=”Standard Normal Distribution, X, 10000 values”, xlab=”Range of random numbers”, ylab=”Density”, breaks=25)
dev.copy(png,”plot2_3_X.png”);
dev.off ();

hist(z2, main=”Standard Normal Distribution, Y, 10000 values”, xlab=”Range of random numbers”, ylab=”Density”, breaks=25)
dev.copy(png,”plot2_3_Y.png”);
dev.off ();


norm <- function(x, mu, sig) {
   return ( (1/ ((2*pi)^(1/2) * sig )) * exp ( -((x-mu)/sig)^2 / 2) )
}


##For N(  = 0,   =5)
x1 <- z1[1:500] * 5^(1/2)
y1 <- z2[1:500] * 5^(1/2)

cat(”N(  = 0,   =5) for 500 values\n”)
cat(”Sample's Mean, X = ”, mean(x1[1:500]), ”, Y = ”, mean(y1[1:500]), ”\n”)
cat(”Sample's Variance, X = ”, var(x1[1:500]), ”, Y = ”, var(y1[1:500]), ”\n”)
cat(”Sample's Covariance = ”, cov(x1[1:500], y1[1:500]), ”\n\n”)

plot(x1[1:500], y1[1:500], col='black', cex=0.5, main=”2-D N(  = 0,   =5), 500 values”, xlab=”X”, ylab=”Y”)
dev.copy(png,”plot4_1.png”);
dev.off ();

```

```
hist(x1[1:500], main="N(  = 0,   =5), X, 500 values", xlab="Range of random numbers", ylab="
     Density", breaks=20)
points(x1 , 500 * norm(x1, 0, 5) , col='black', cex=0.5)
dev.copy(png,"plot4_1_X.png");
dev.off ();

hist(y1[1:500], main="N(  = 0,   =5), Y, 500 values", xlab="Range of random numbers", ylab="
     Density", breaks=20)
points(y1 , 500 * norm(y1, 0, 5) , col='black', cex=0.5)
dev.copy(png,"plot4_1_Y.png");
dev.off ();


##For N(  = 5,   =5)



x2 <- x1 + 5
y2 <- y1 + 5

cat("N(  = 5,   =5) for 500 values\n")
cat("Sample's Mean, X = ", mean(x2[1:500]), ", Y = ", mean(y2[1:500]), "\n")
cat("Sample's Variance, X = ", var(x2[1:500]), ", Y = ", var(y2[1:500]), "\n")
cat("Sample's Covariance = ", cov(x2[1:500], y2[1:500]), "\n\n")

plot(x2[1:500], y2[1:500], col='black', cex=0.5, main="2-D N(  = 5,   =5), 500 values", xlab="
     X", ylab="Y")
dev.copy(png,"plot4_2.png");
dev.off ();

hist(x2[1:500], main="N(  = 5,   =5), X, 500 values", xlab="Range of random numbers", ylab="
     Density", breaks=20)
points(x2 , 500 * norm(x2, 5, 5) , col='black', cex=0.5)
dev.copy(png,"plot4_2_X.png");
dev.off ();

hist(y2[1:500], main="N(  = 5,   =5), Y, 500 values", xlab="Range of random numbers", ylab="
     Density", breaks=20)
points(y2 , 500 * norm(y2, 5, 5) , col='black', cex=0.5)
dev.copy(png,"plot4_2_Y.png");
dev.off ();
rm(list = ls())
```

3 Question 1

Box-Muller Method

For 100 values

Sample's Mean, X = -0.1596442 , Y = 0.08031238

Sample's Variance, X = 1.012568 , Y = 0.8934606

Sample's Covariance = -0.05055587

For 500 values

Sample's Mean, X = -0.09864711 , Y = 0.03518648

Sample's Variance, X = 1.059539 , Y = 0.8675763

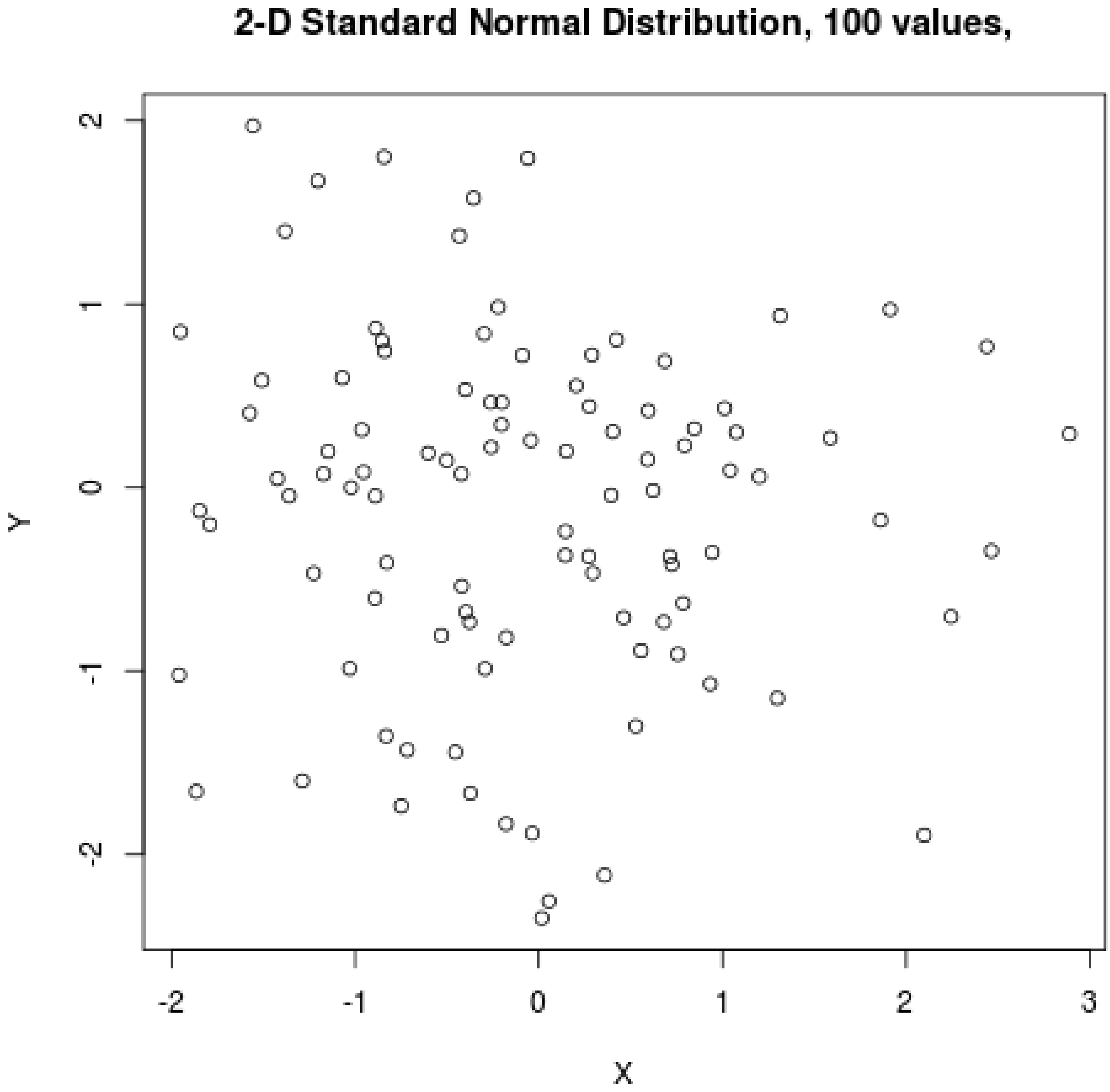
Sample's Covariance = -0.05837014

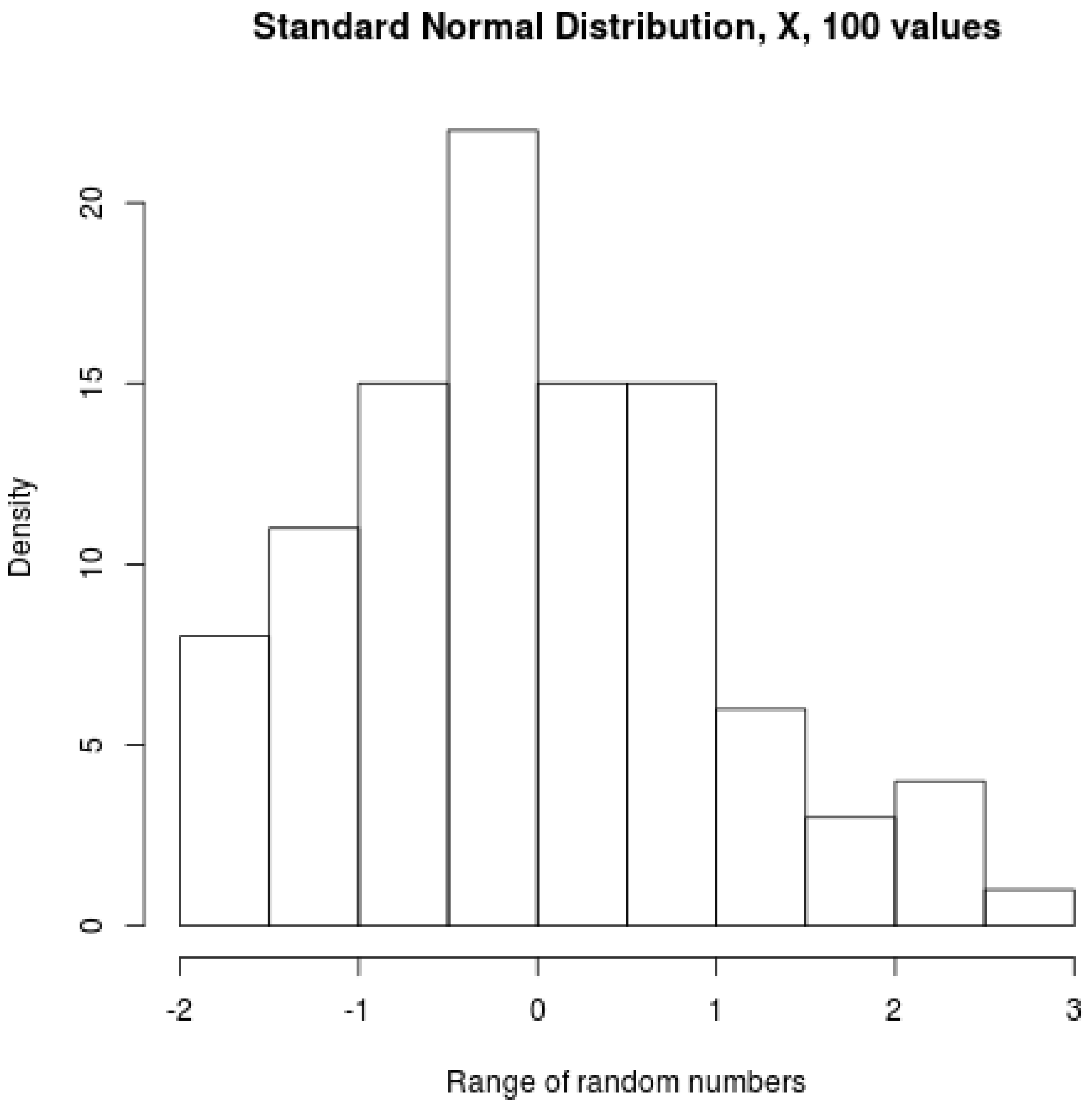
For 10000 values

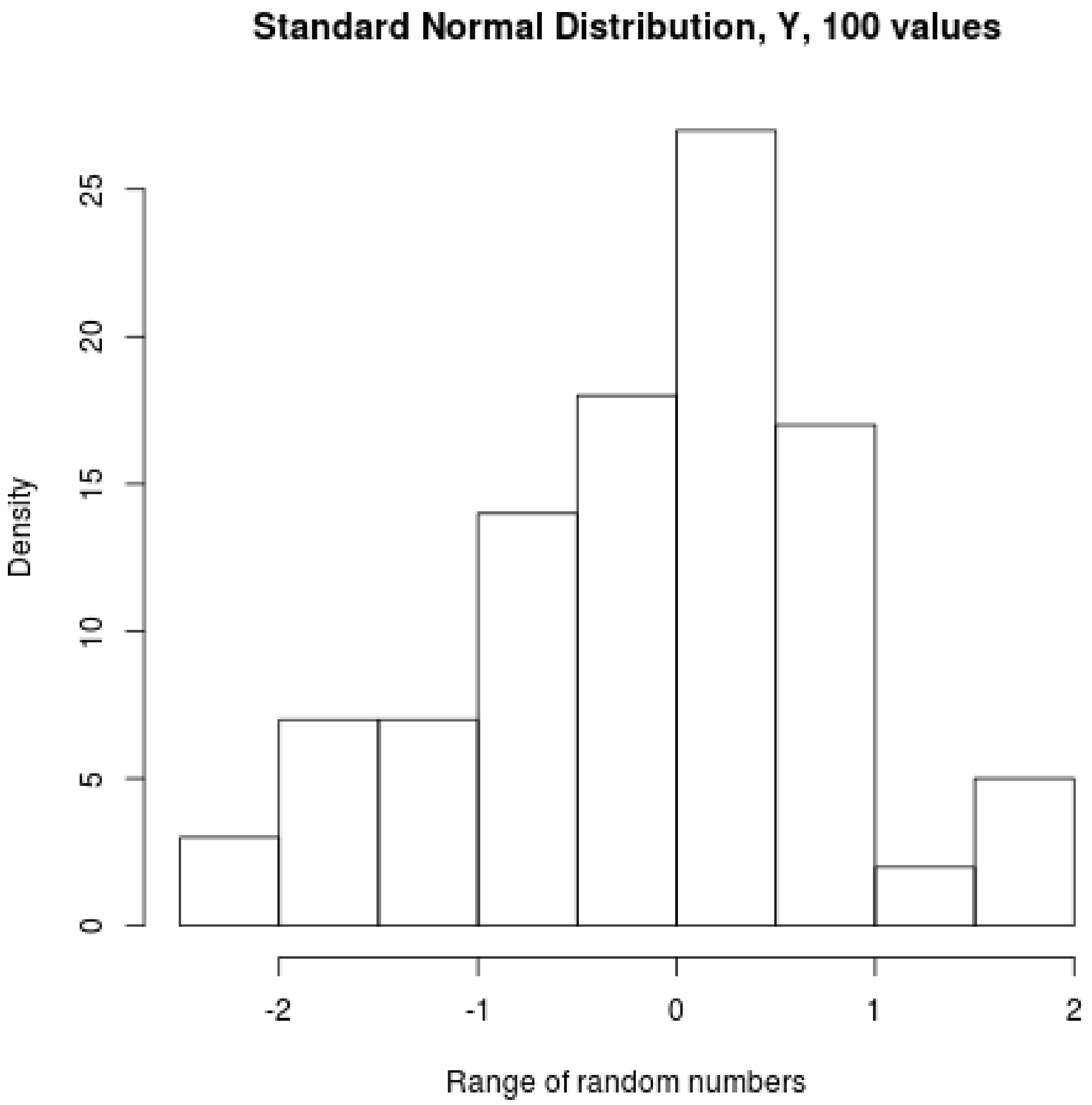
Sample's Mean, X = -0.02360723 , Y = 0.01601492

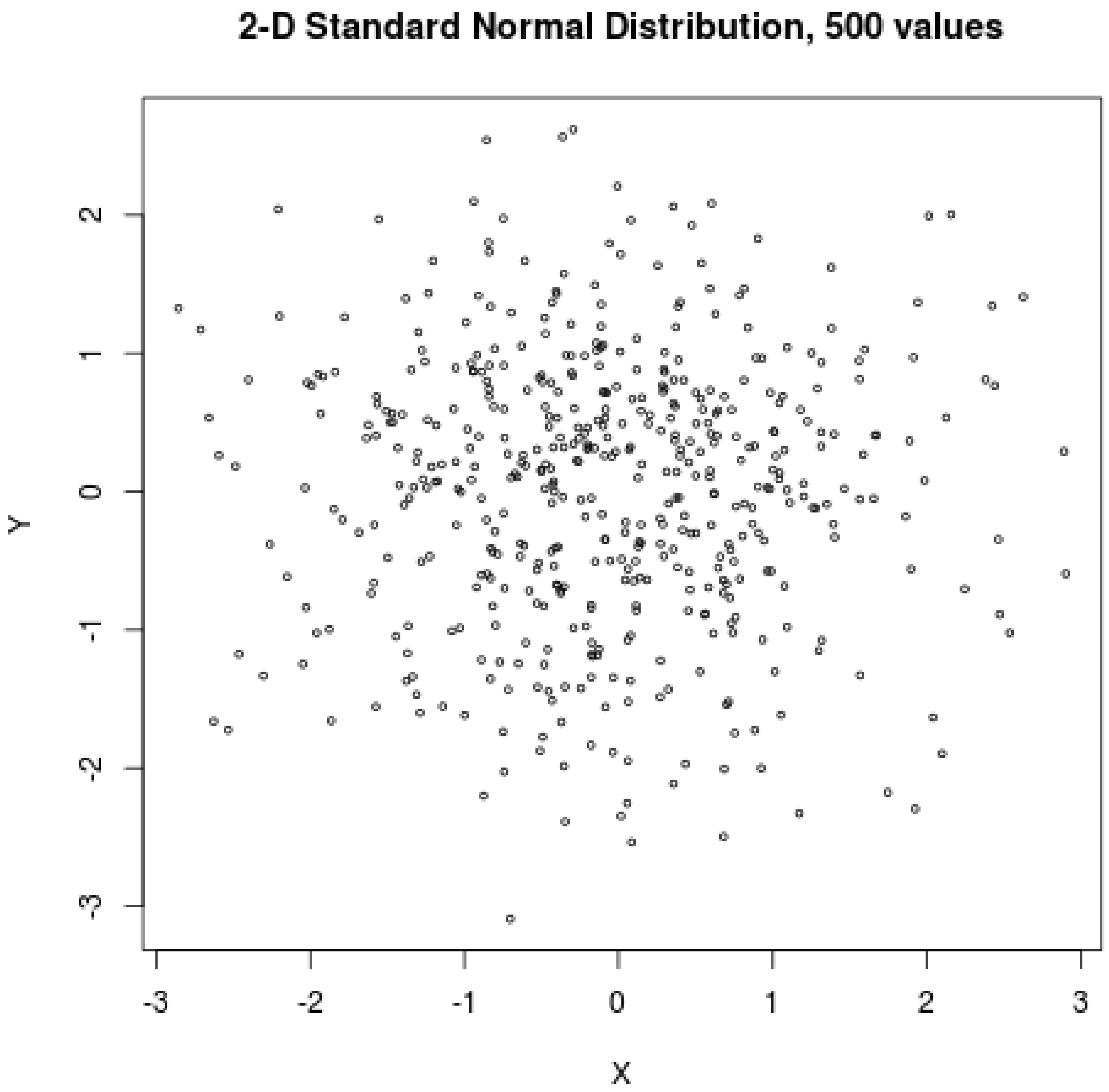
Sample's Variance, X = 0.9678944 , Y = 0.9818059

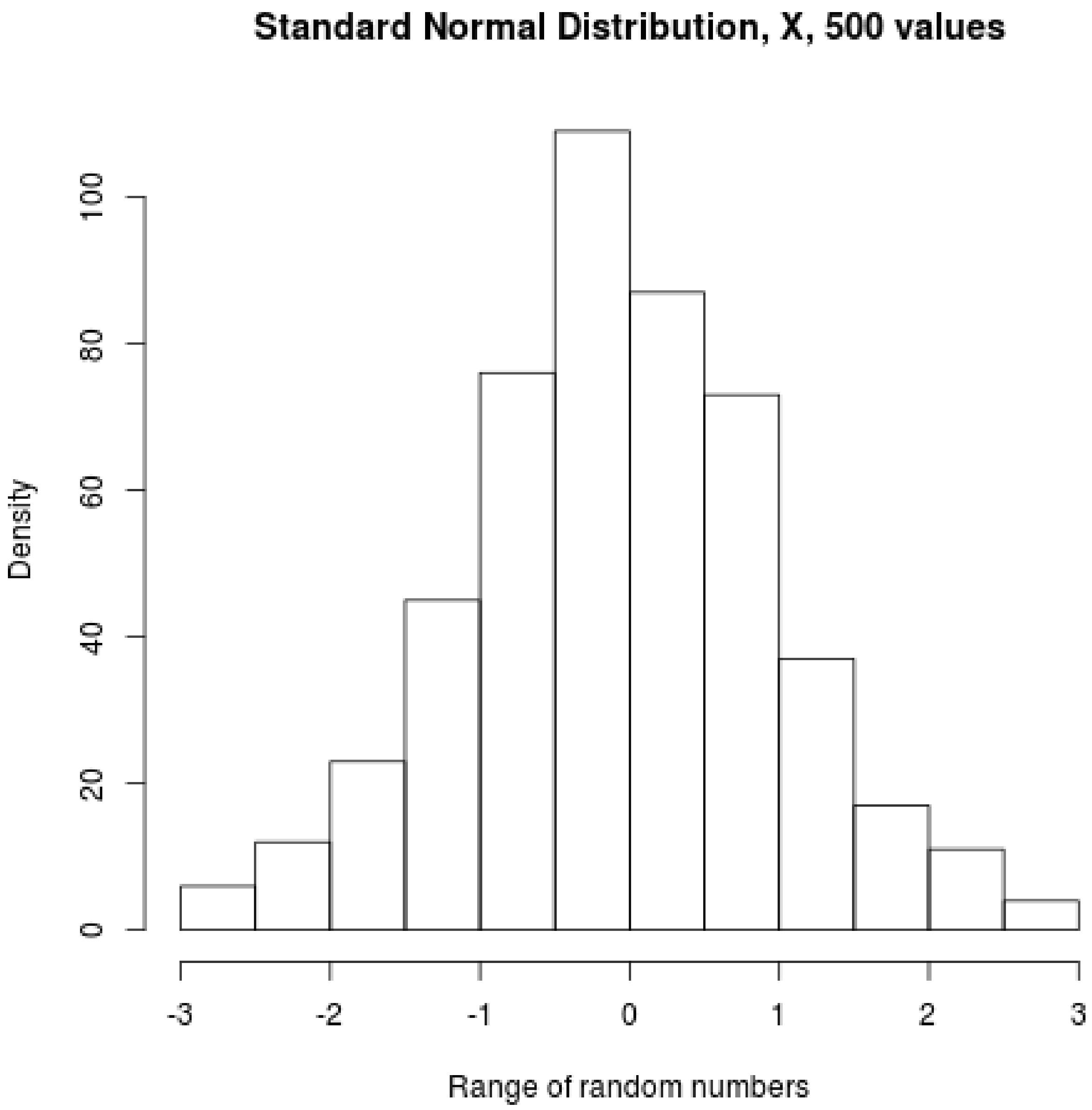
Sample's Covariance = -0.01037898

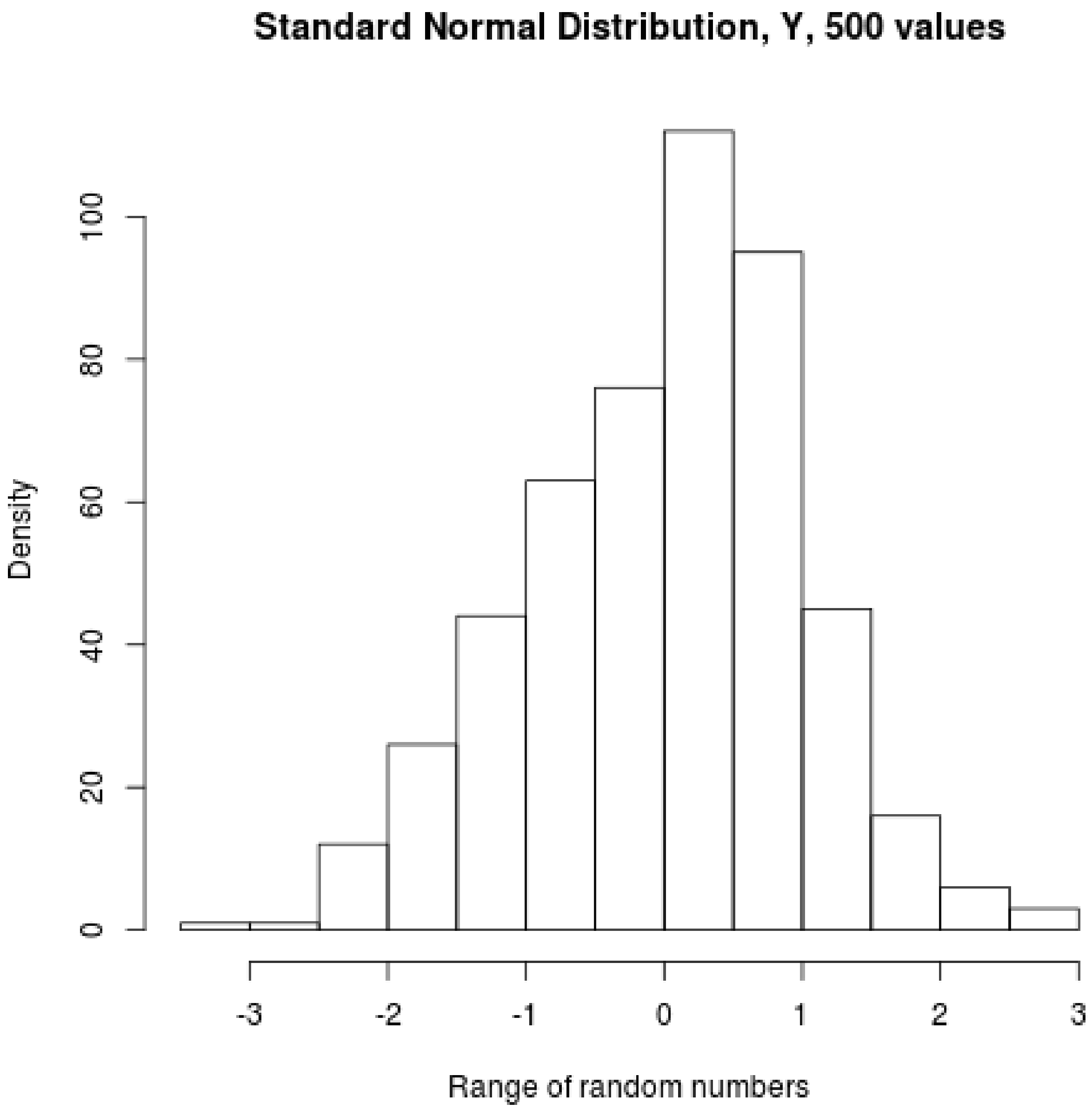

2-D Standard Normal Distribution, 100 values,
Y
2
1
0
-1
-2
X
-2
-1
0
1
2
3

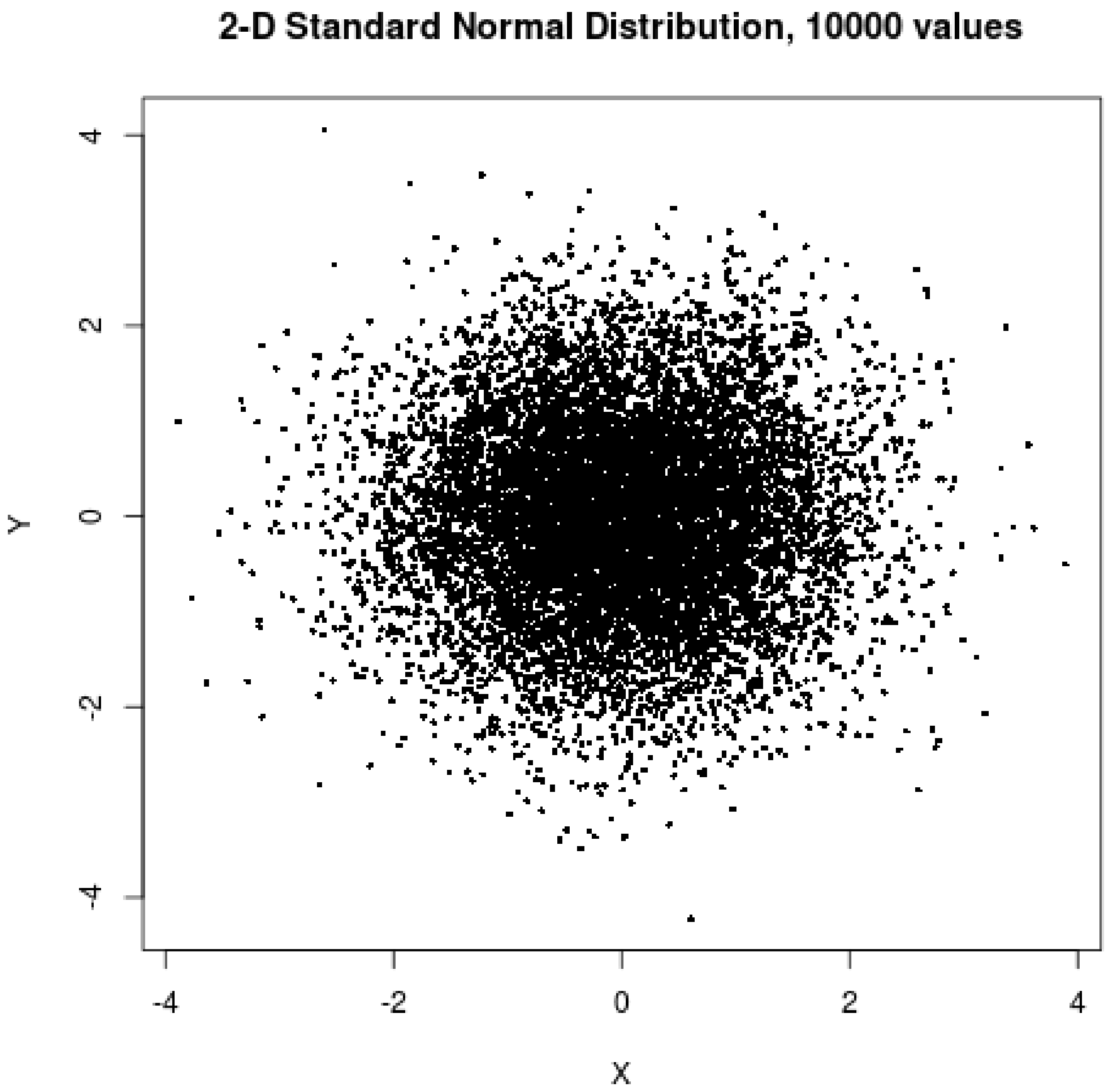

Standard Normal Distribution, X, 100 values
Density
0
5
10
15
20
-2
-1
0
1
2
3
Range of random numbers

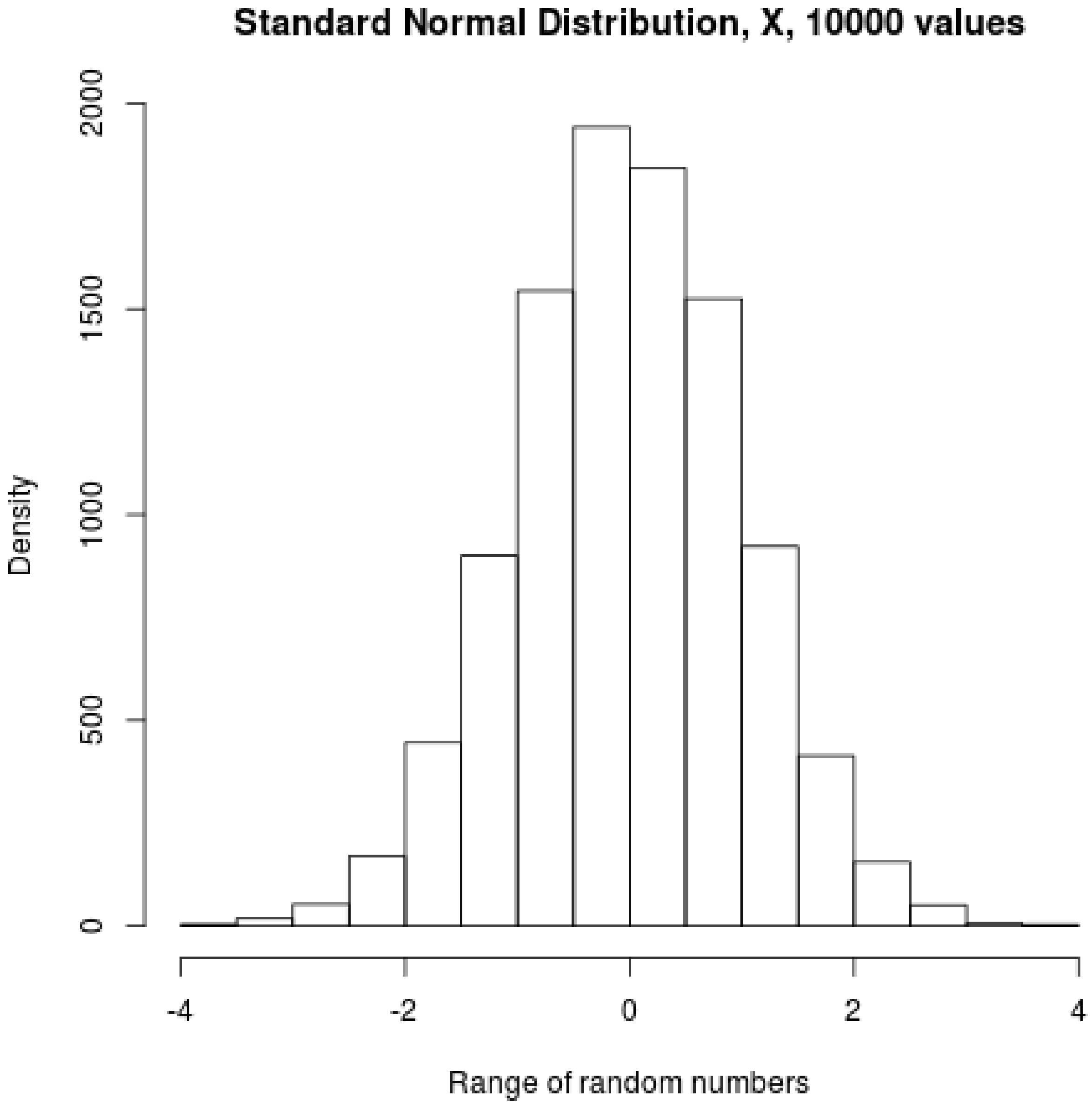

Standard Normal Distribution, Y, 100 values
Density
0
5
10
15
20
25
-2
-1
0
1
2
Range of random numbers

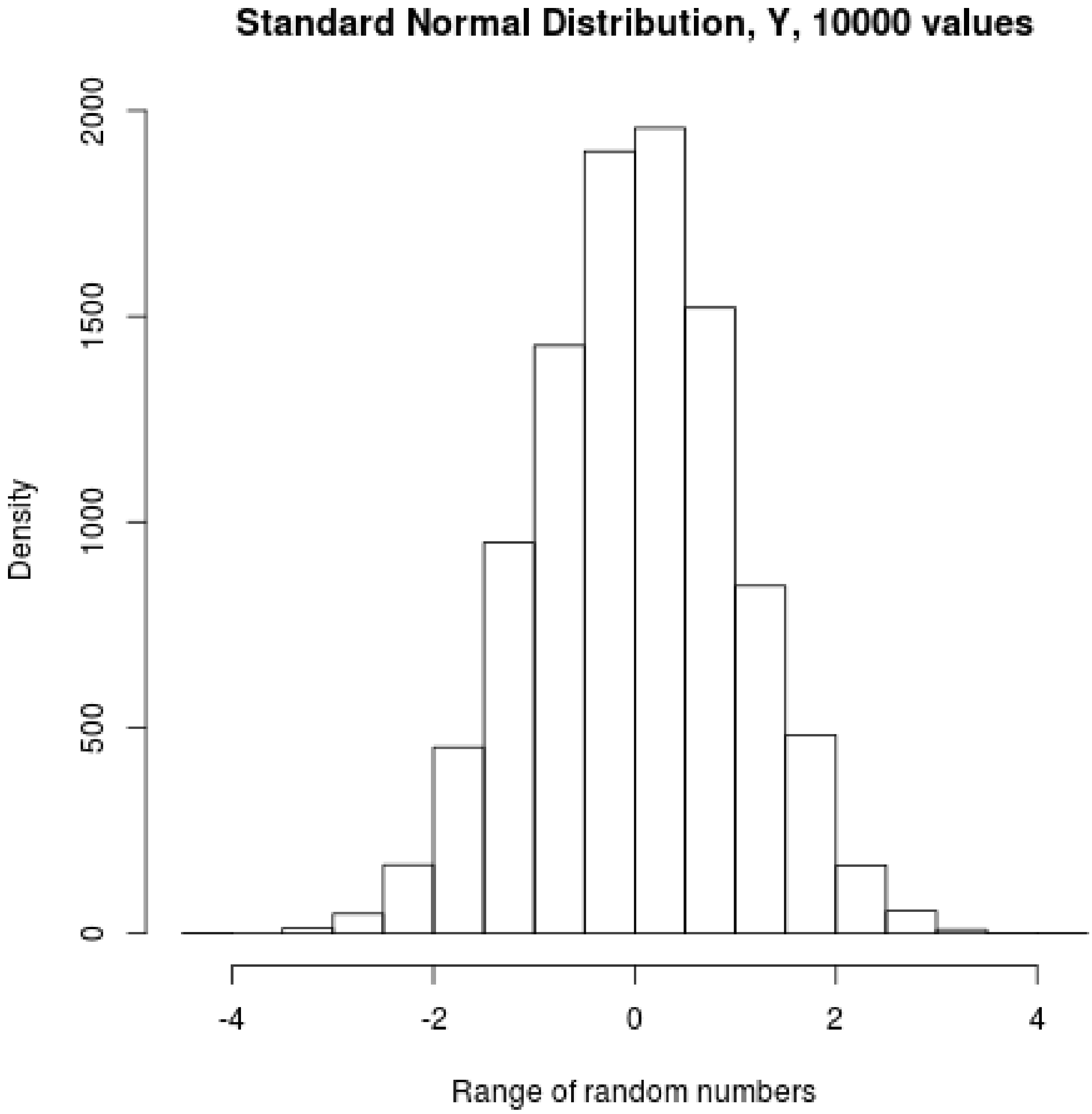

2-D Standard Normal Distribution, 500 values
Y
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3
X


Standard Normal Distribution, X, 500 values
Density
0
20
40
60
80
100
-3
-2
-1
0
1
2
3
Range of random numbers


Standard Normal Distribution, Y, 500 values
Density
0
20
40
60
80
100
-3
-2
-1
0
1
2
3
Range of random numbers


2-D Standard Normal Distribution, 10000 values
Y
4
2
0
-2
-4
X
-4
-2
0
2
4


Standard Normal Distribution, X, 10000 values
Density
0
500
1000
1500
2000
-4
-2
0
2
4
Range of random numbers


Standard Normal Distribution, Y, 10000 values
Density
0
500
1000
1500
2000
-4
-2
0
2
4
Range of random numbers

Marsaglia-Bray Method

For 100 values

Sample’s Mean, X = 0.02650457 , Y = 0.08917513

Sample’s Variance, X = 1.036414 , Y = 0.9381411

Sample’s Covariance = 0.03333995

For 500 values

Sample’s Mean, X = 0.01449956 , Y = 0.02634573

Sample’s Variance, X = 0.9660087 , Y = 0.8949109

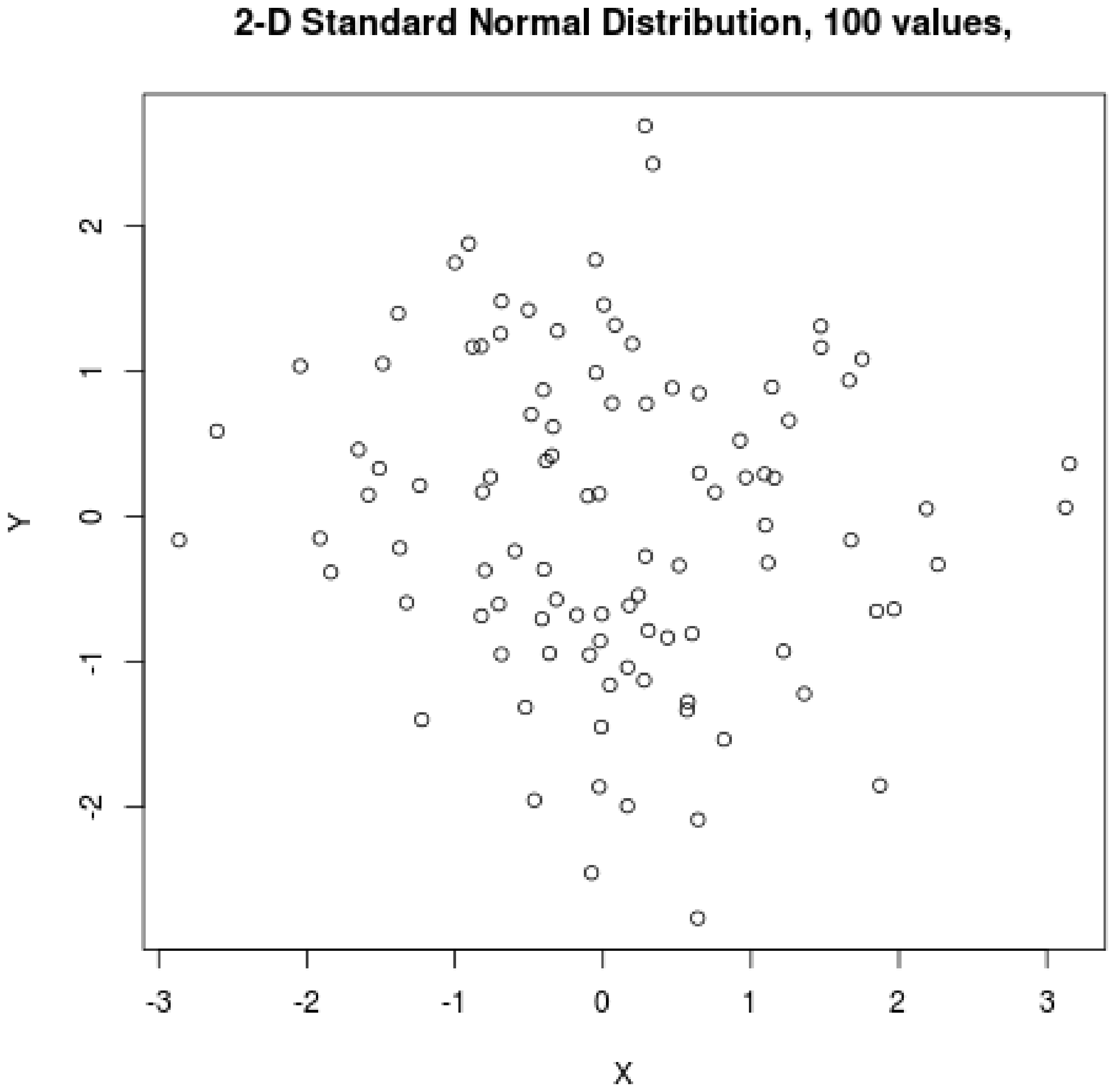
Sample’s Covariance = -0.1012995

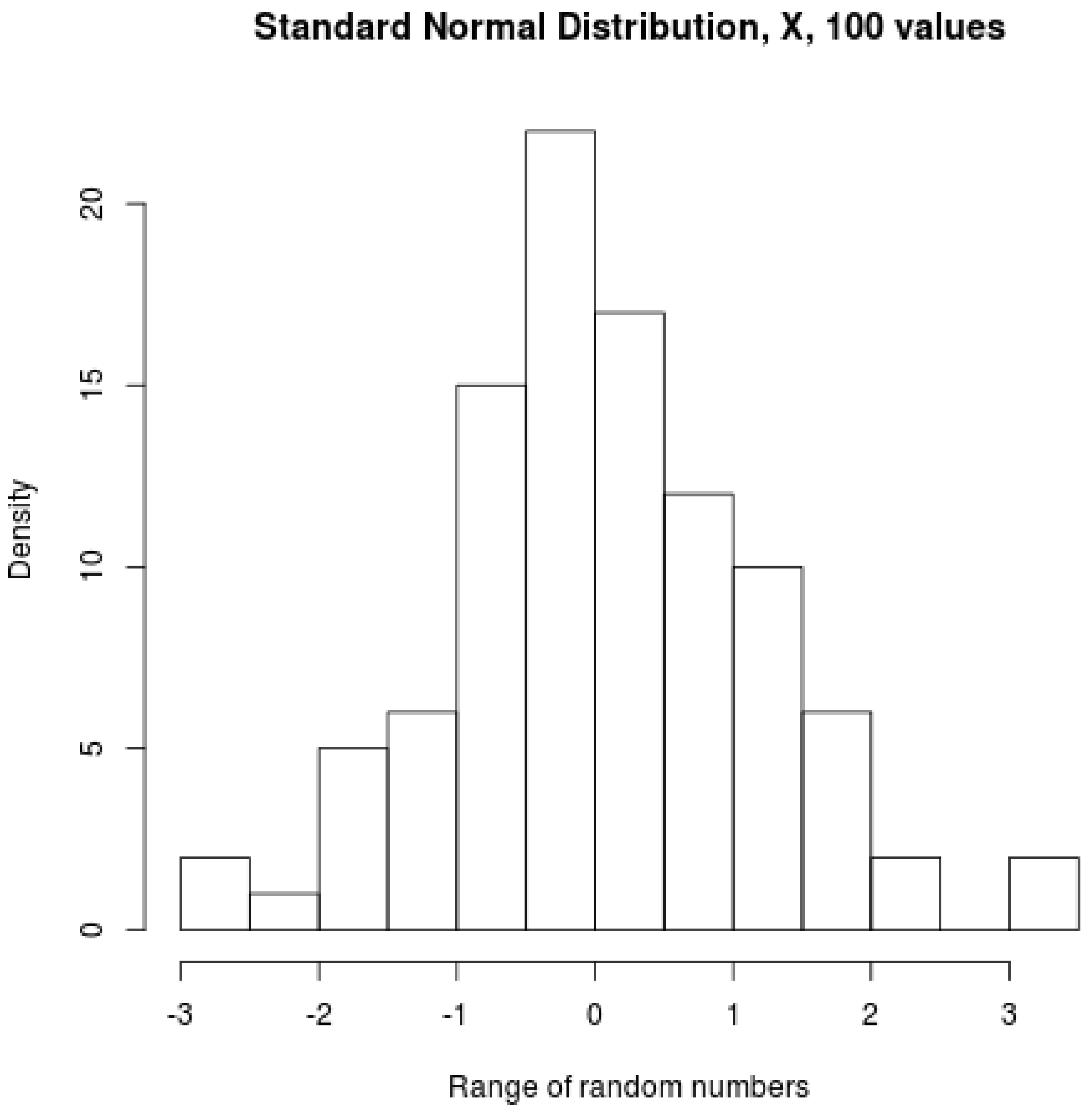
For 10000 values

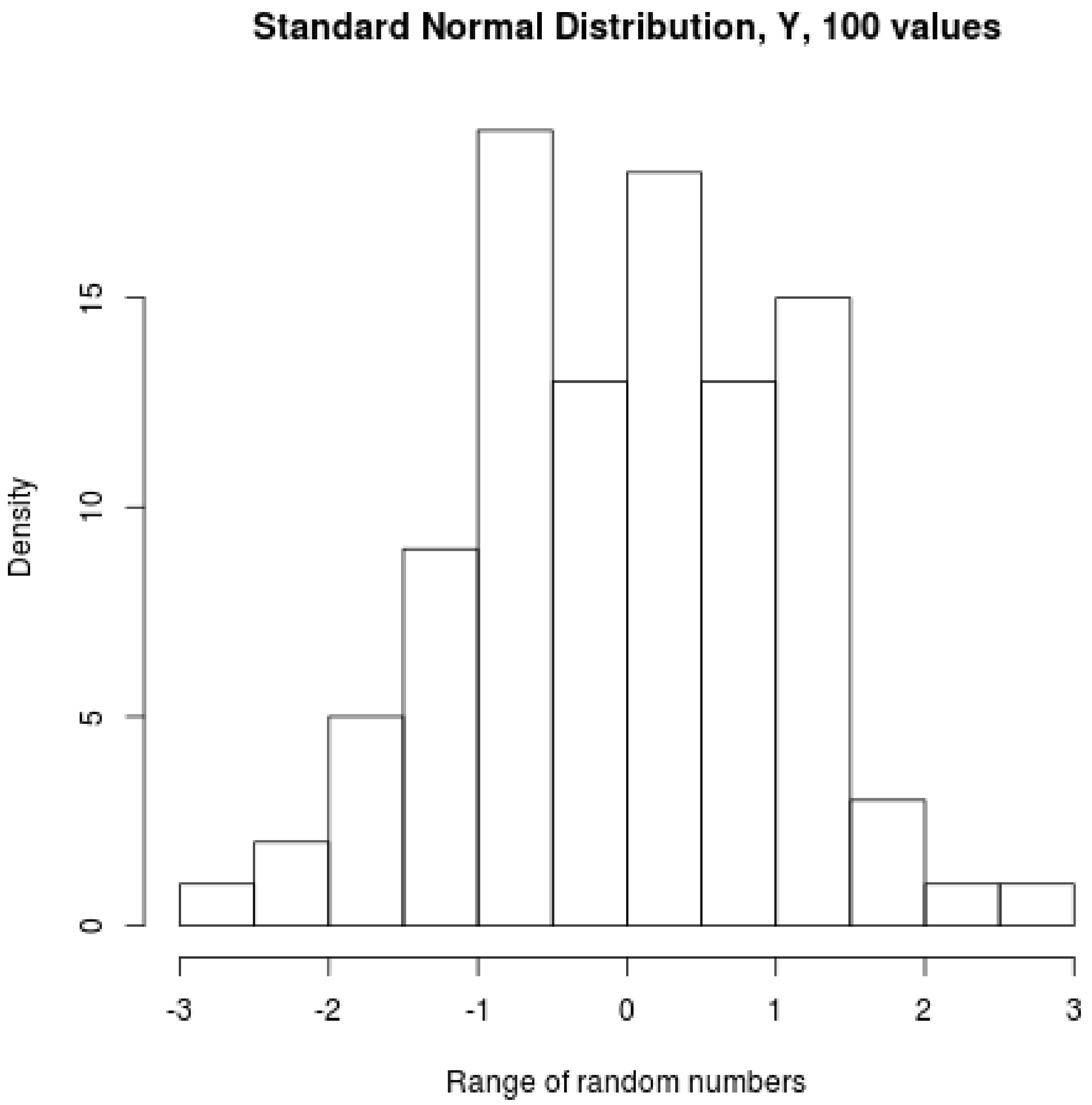
Sample’s Mean, X = -0.002328363 , Y = 0.09431695

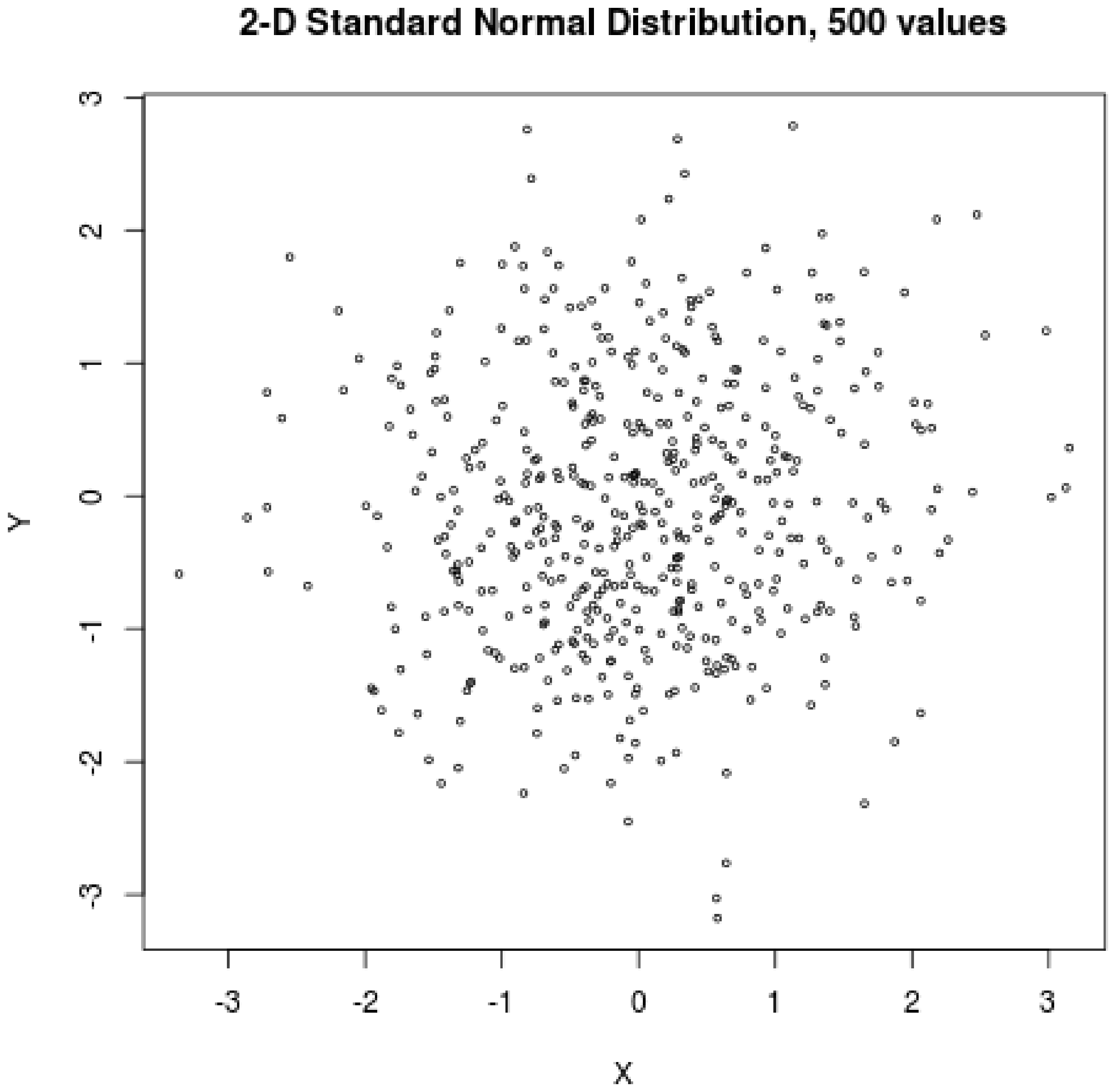
Sample’s Variance, X = 0.9546331 , Y = 0.9437196

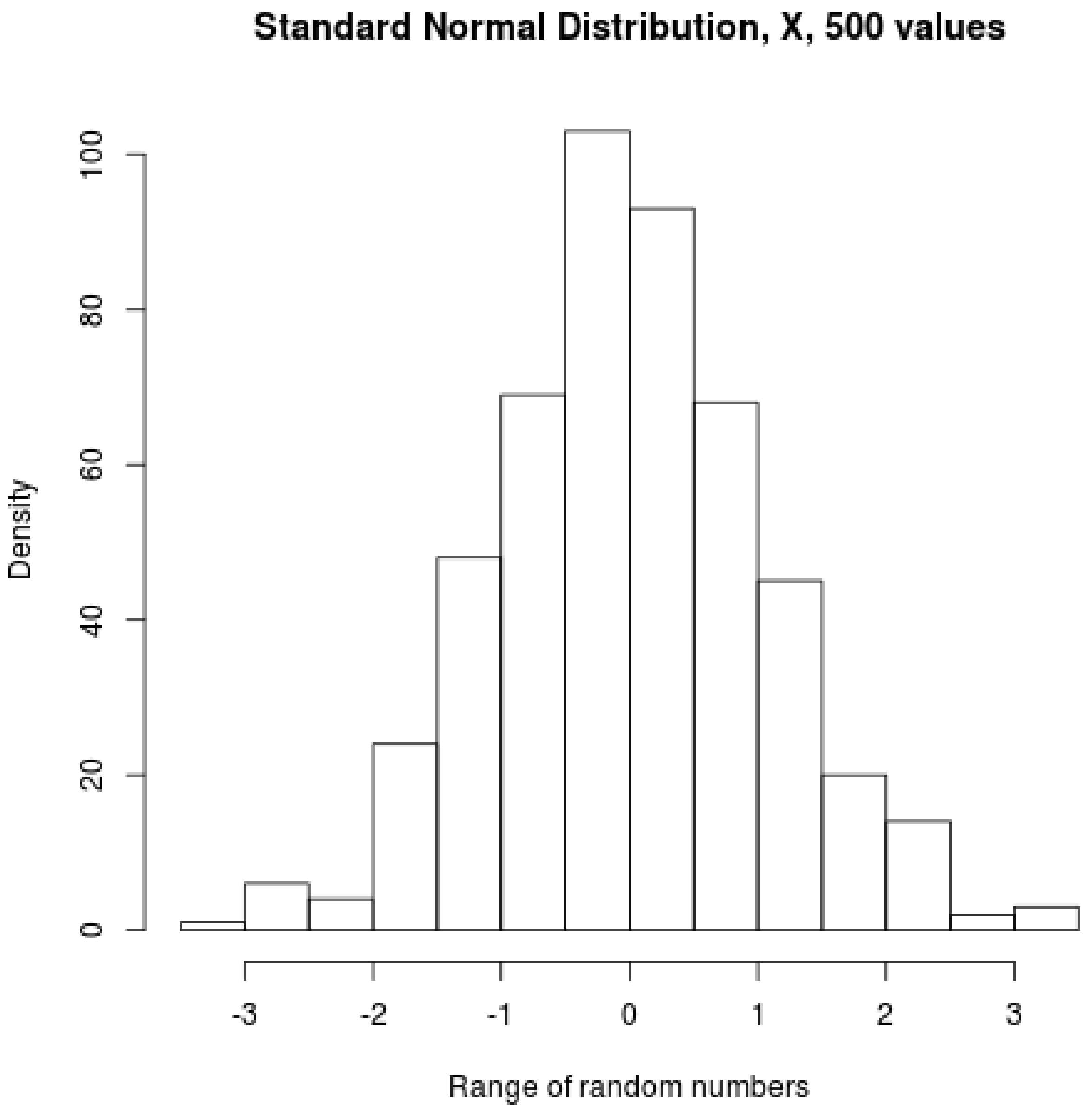
Sample’s Covariance = -0.03138388

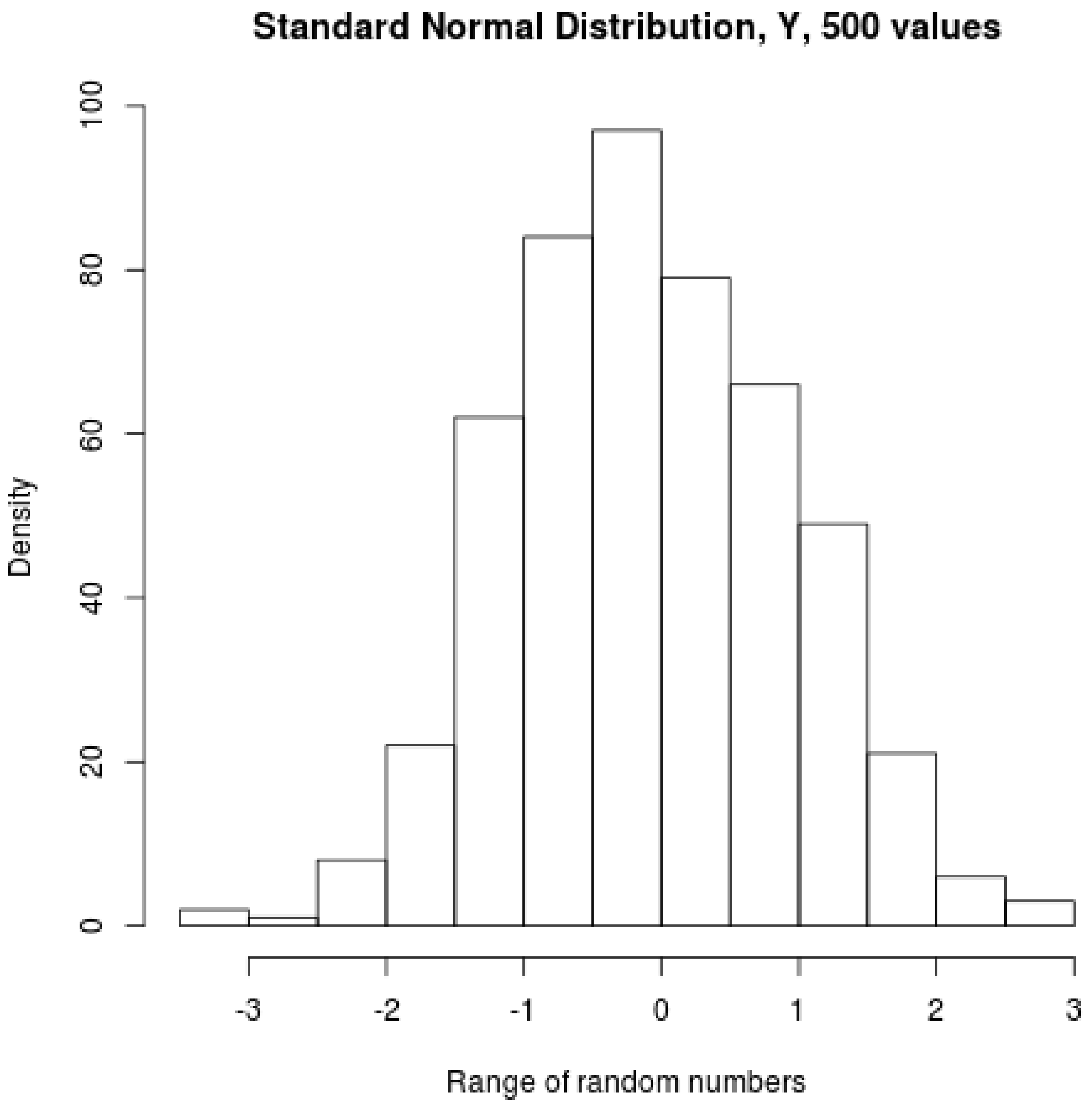

2-D Standard Normal Distribution, 100 values,
Y
2
1
0
-1
-2
-3
-2
-1
0
1
2
3
X

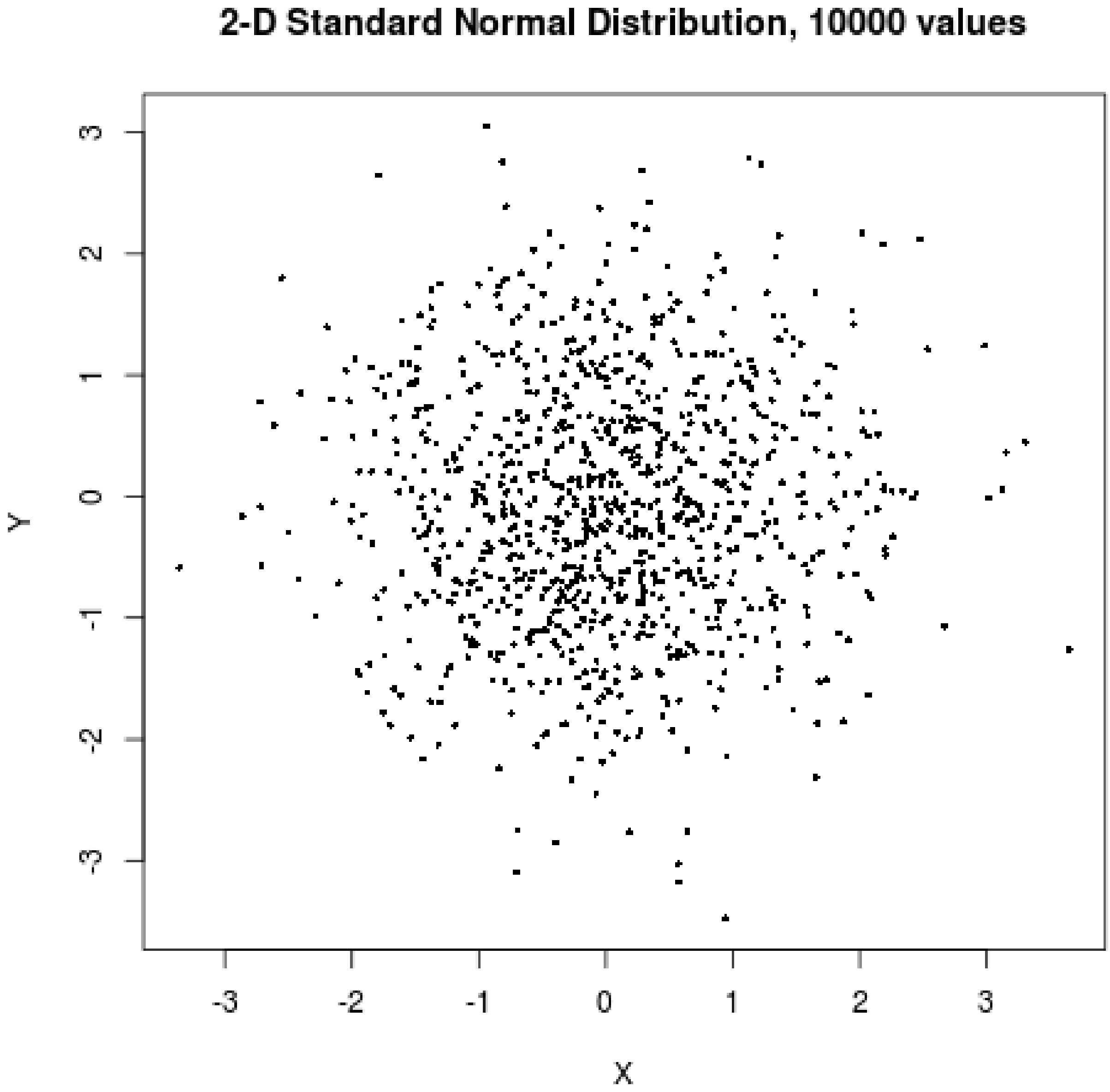

Standard Normal Distribution, X, 100 values
Density
0
5
10
15
20
-3
-2
-1
0
1
2
3
Range of random numbers

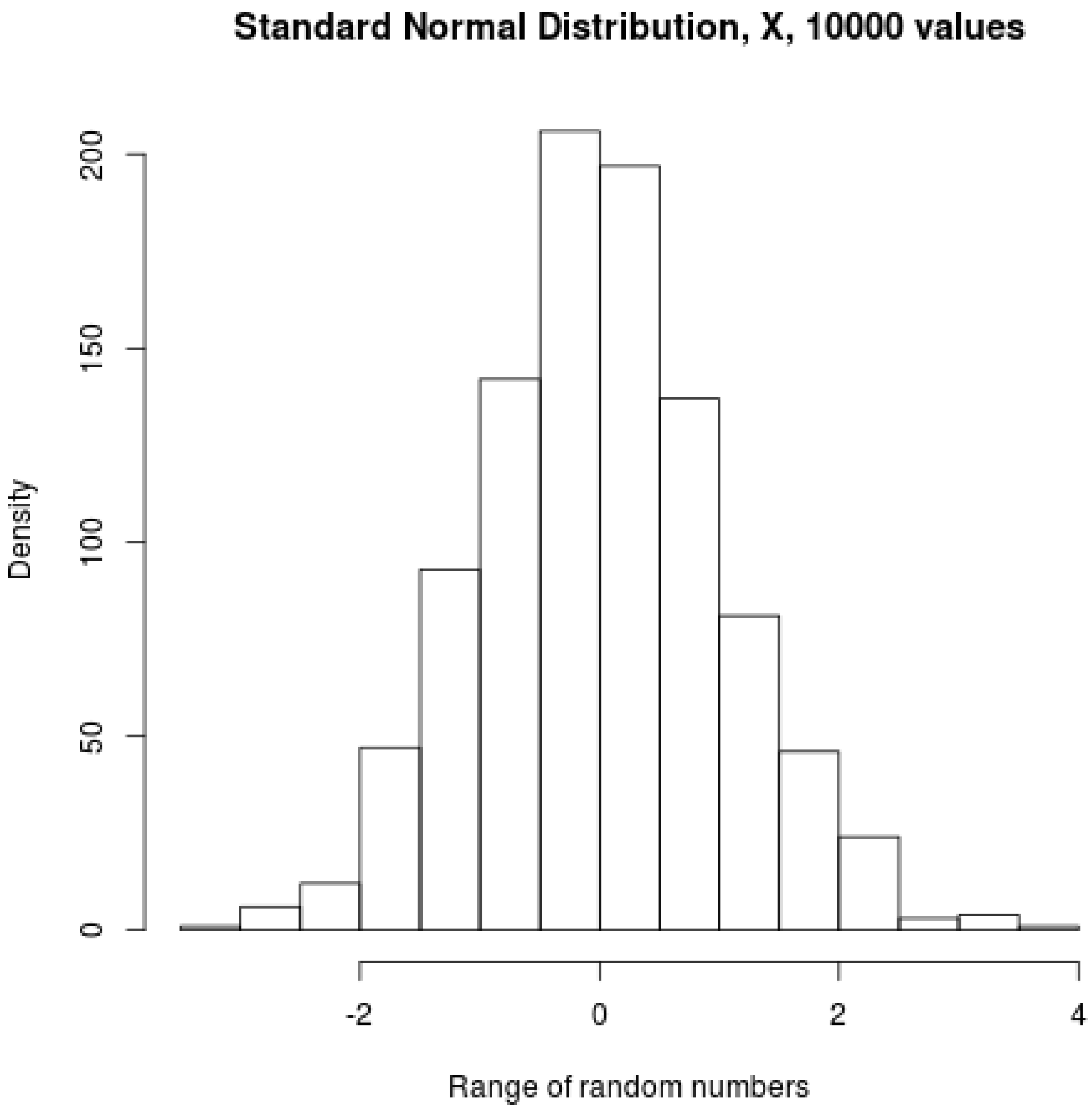

Standard Normal Distribution, Y, 100 values
Density
0
5
10
15
-3
-2
-1
0
1
2
3
Range of random numbers

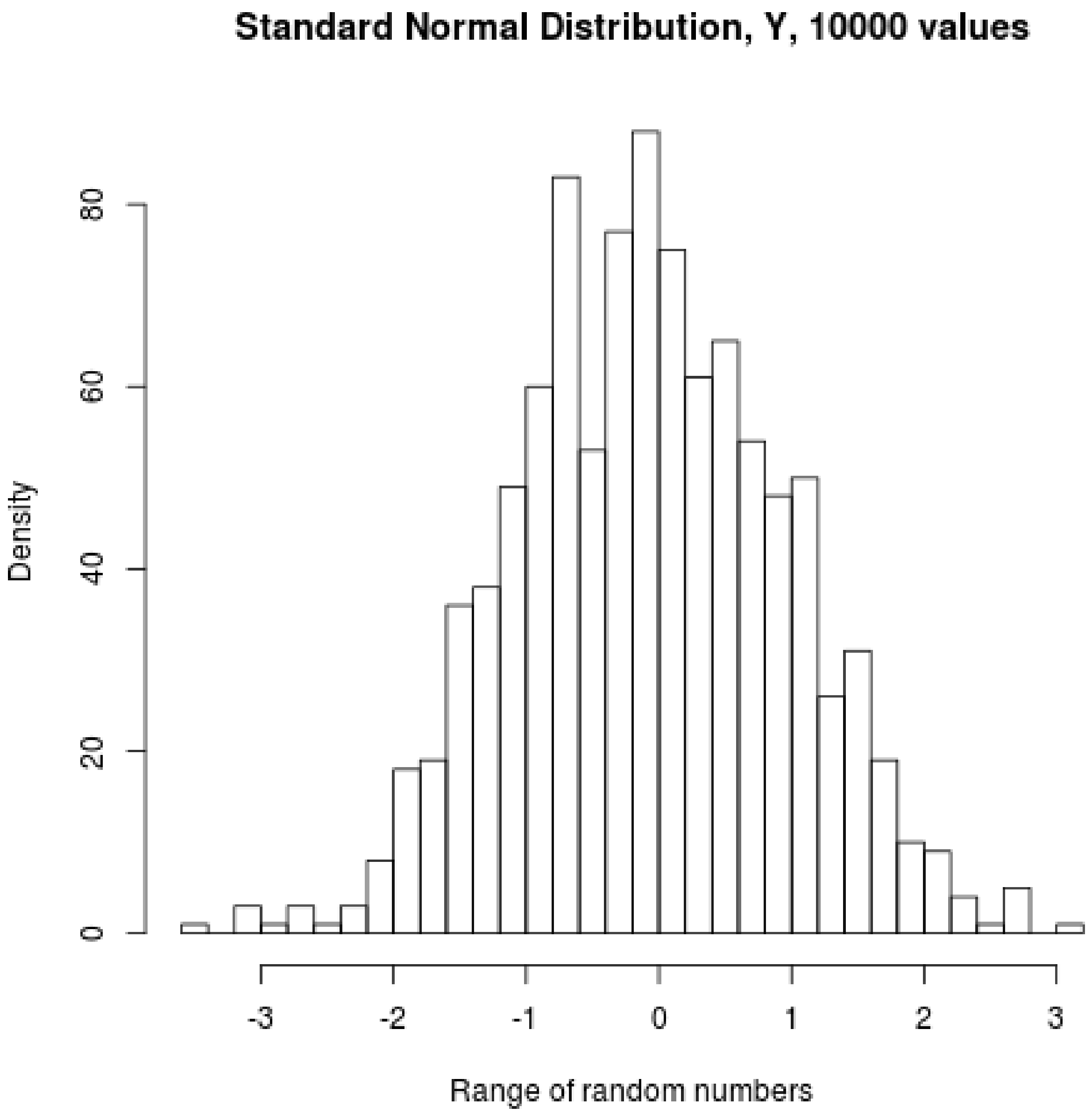

2-D Standard Normal Distribution, 500 values
Y
X
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3


Standard Normal Distribution, X, 500 values
Density
0
20
40
60
80
100
-3
-2
-1
0
1
2
3
Range of random numbers


Standard Normal Distribution, Y, 500 values
Density
0
20
40
60
80
100
-3
-2
-1
0
1
2
3
Range of random numbers


2-D Standard Normal Distribution, 10000 values
Y
3
2
1
0
-1
-2
-3
X
-3
-2
-1
0
1
2
3

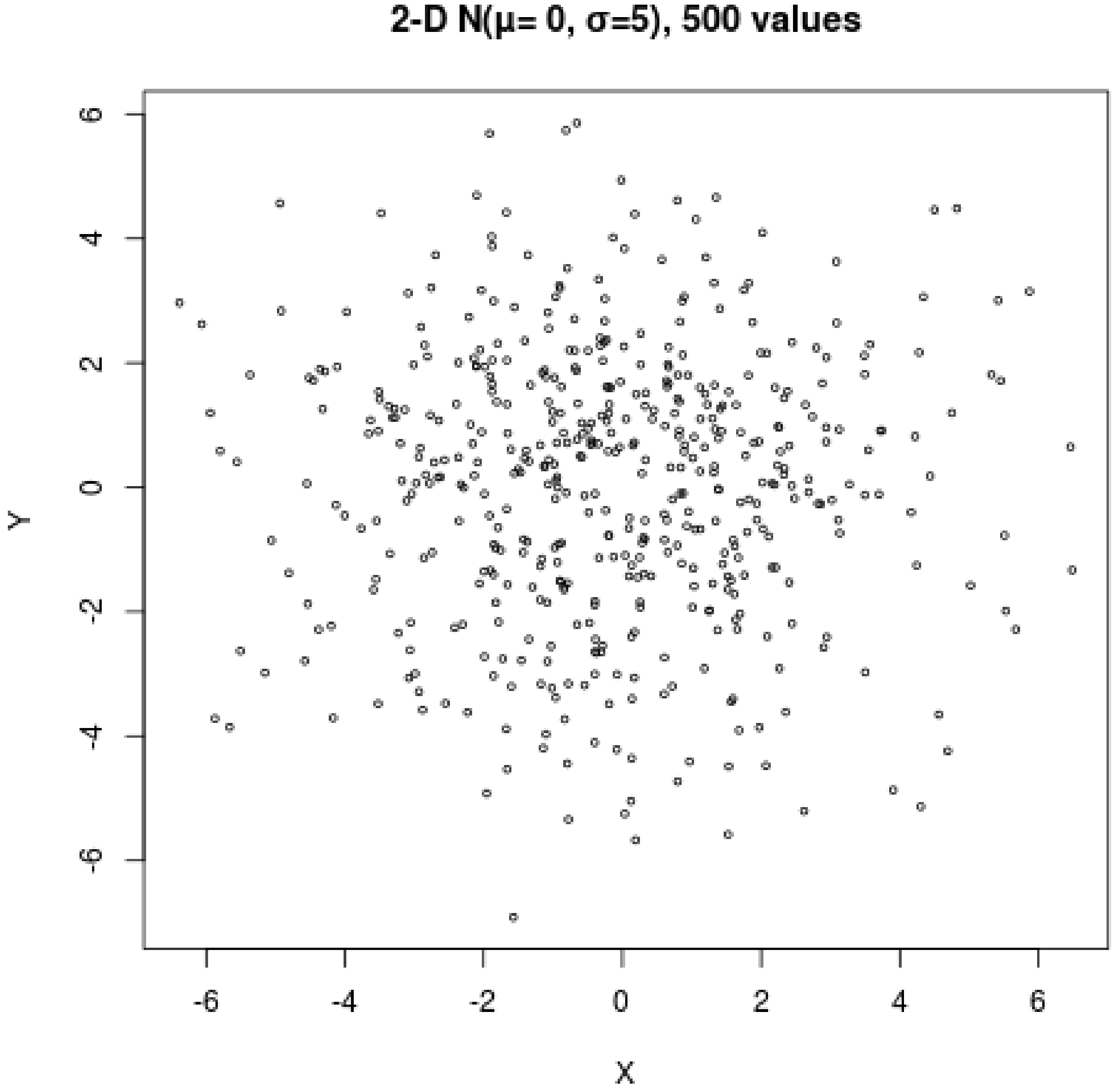

Standard Normal Distribution, X, 10000 values
Density
0
50
100
150
200
-2
0
2
4
Range of random numbers

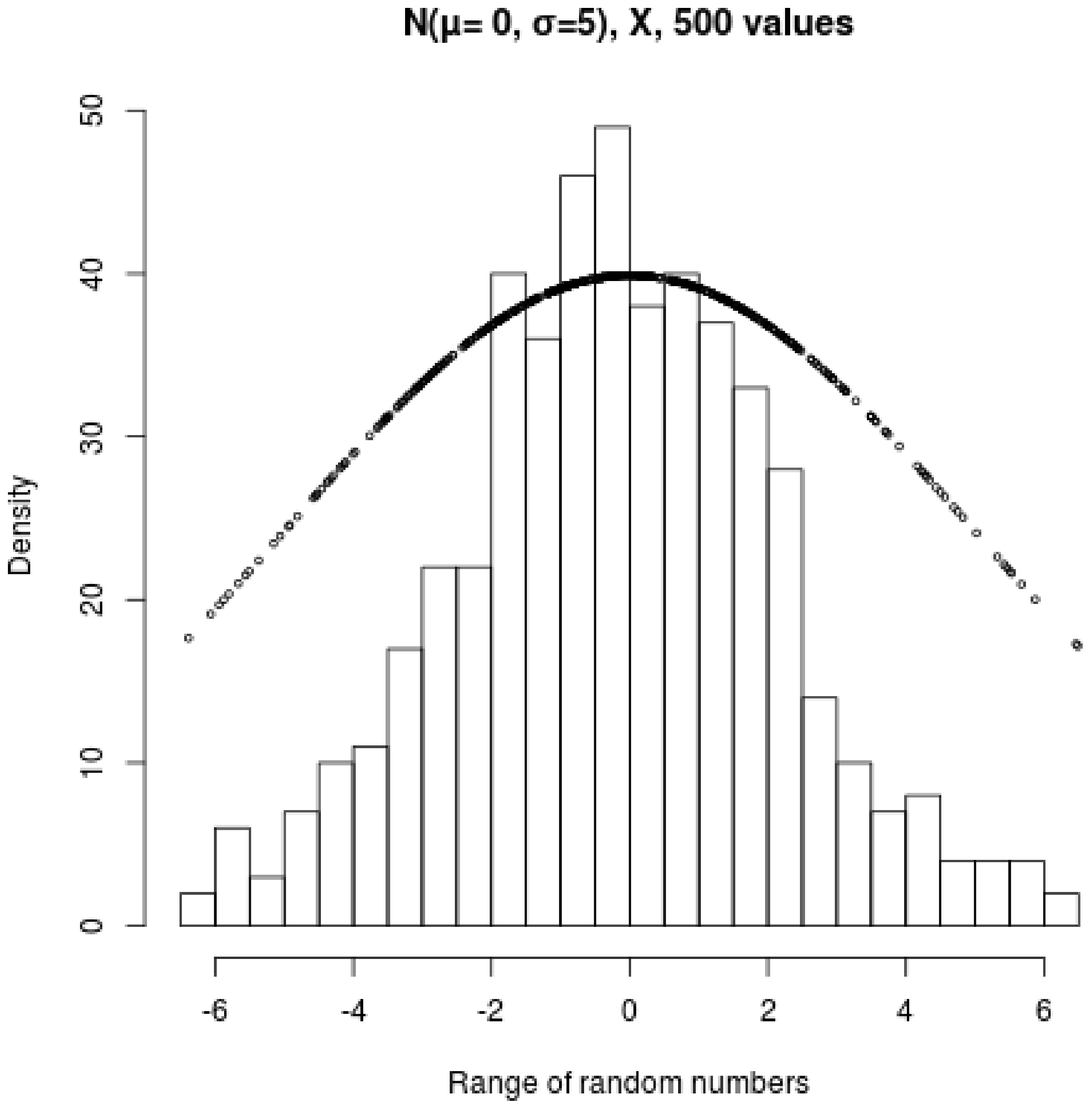

Standard Normal Distribution, Y, 10000 values
Density
0
20
40
60
80
-3
-2
-1
0
1
2
3
Range of random numbers

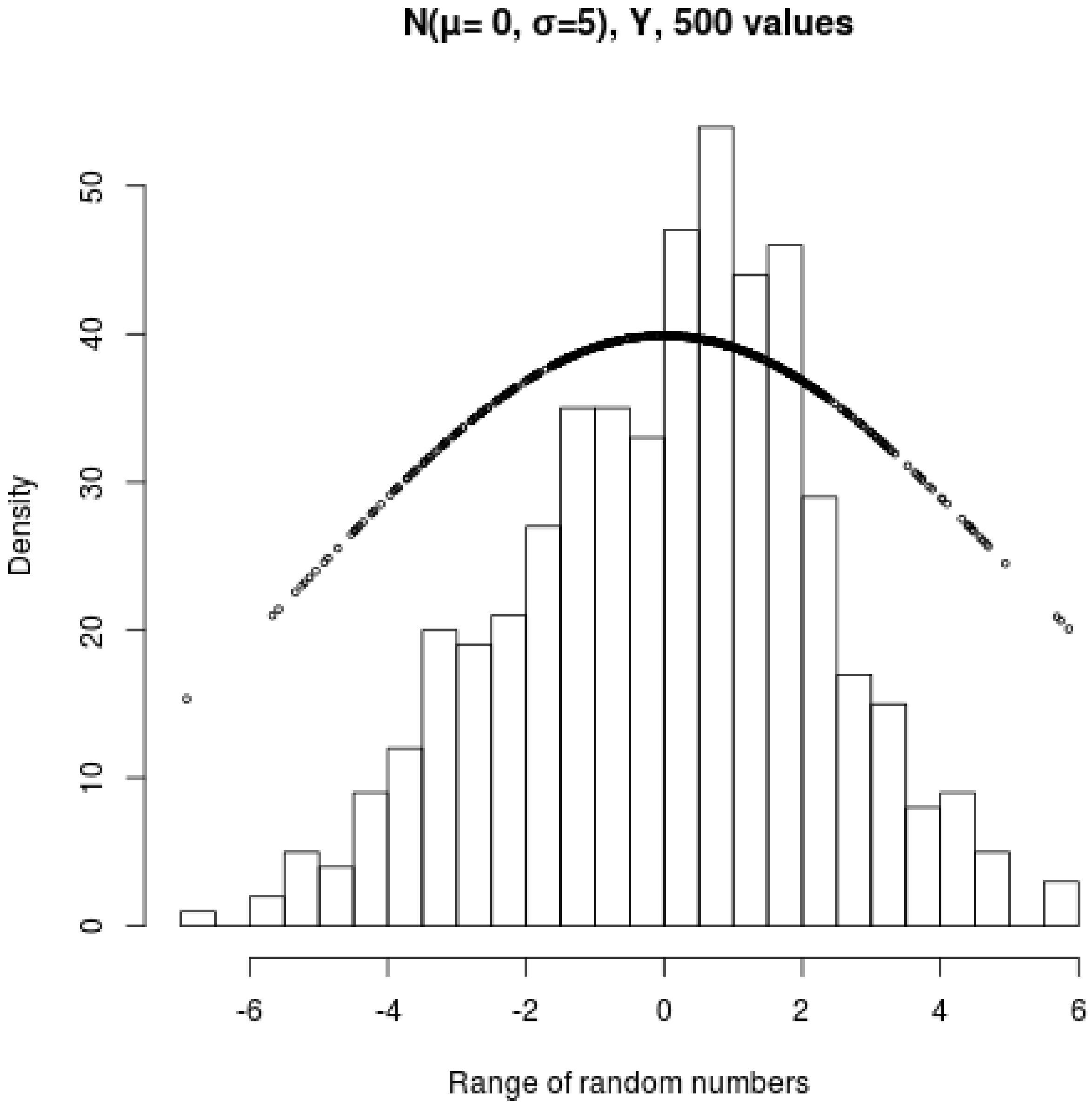
4 Question 2

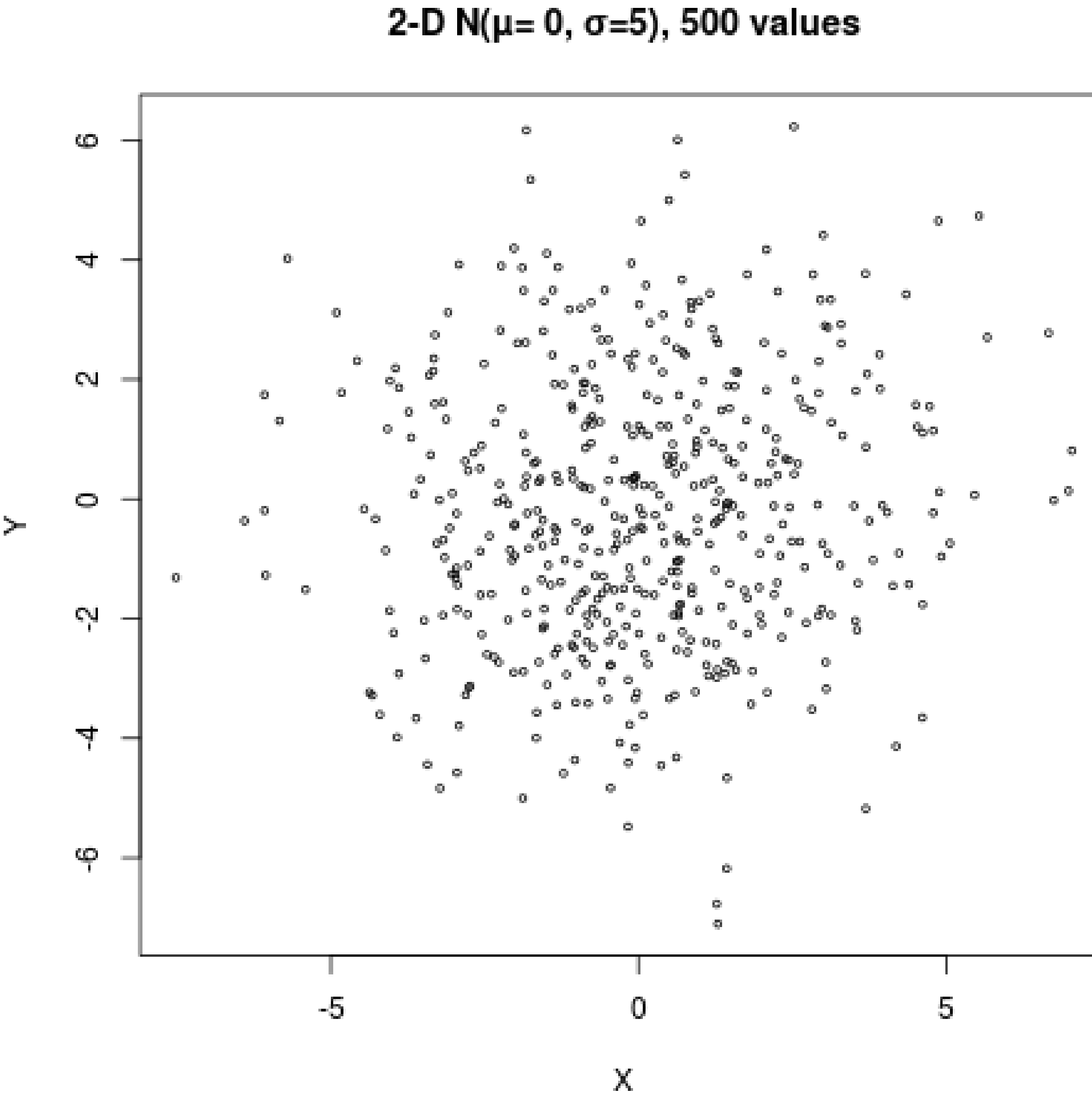
$N(\mu = 0, \sigma = 5)$ for 500 values

By Box-Muller

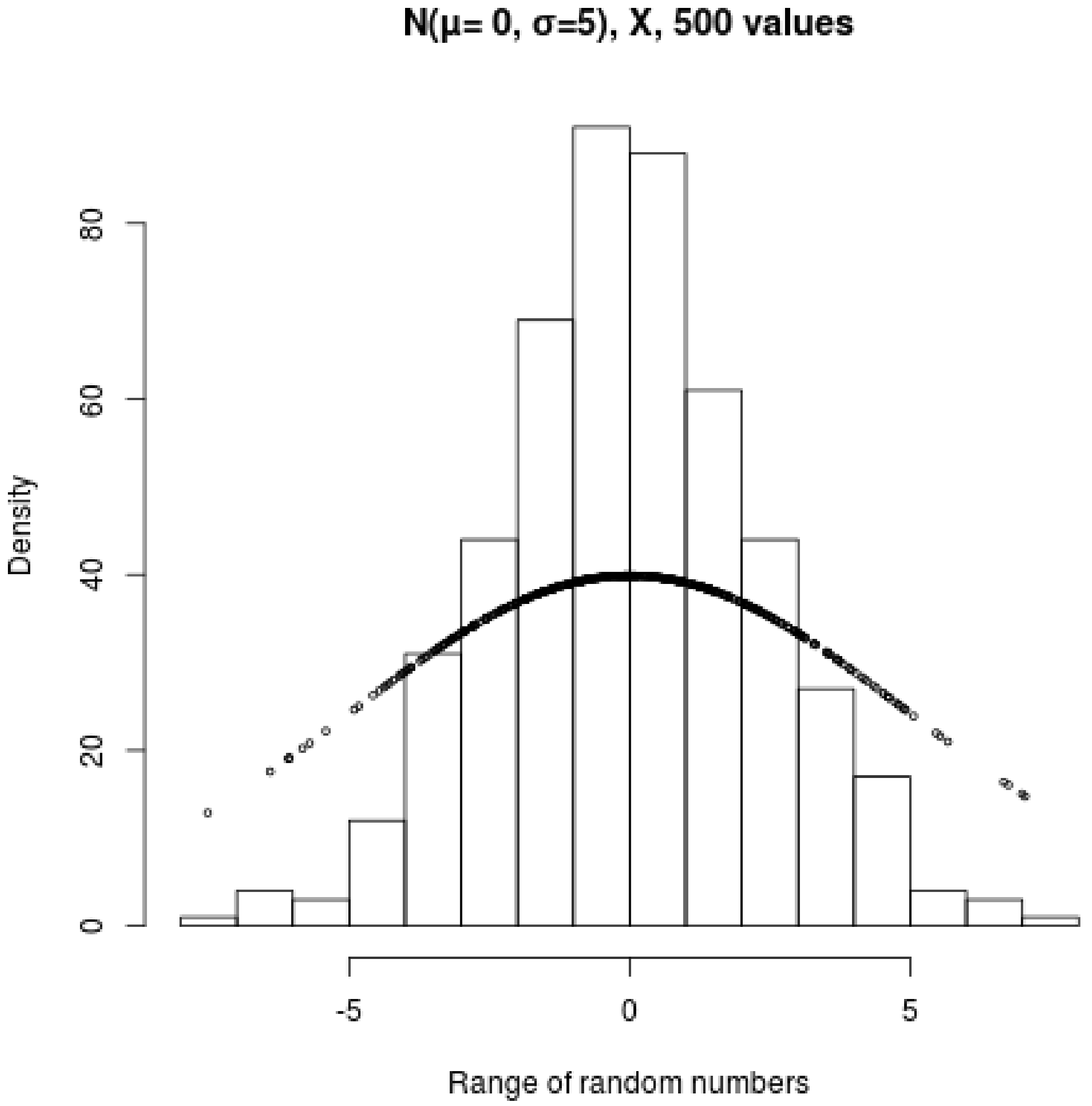


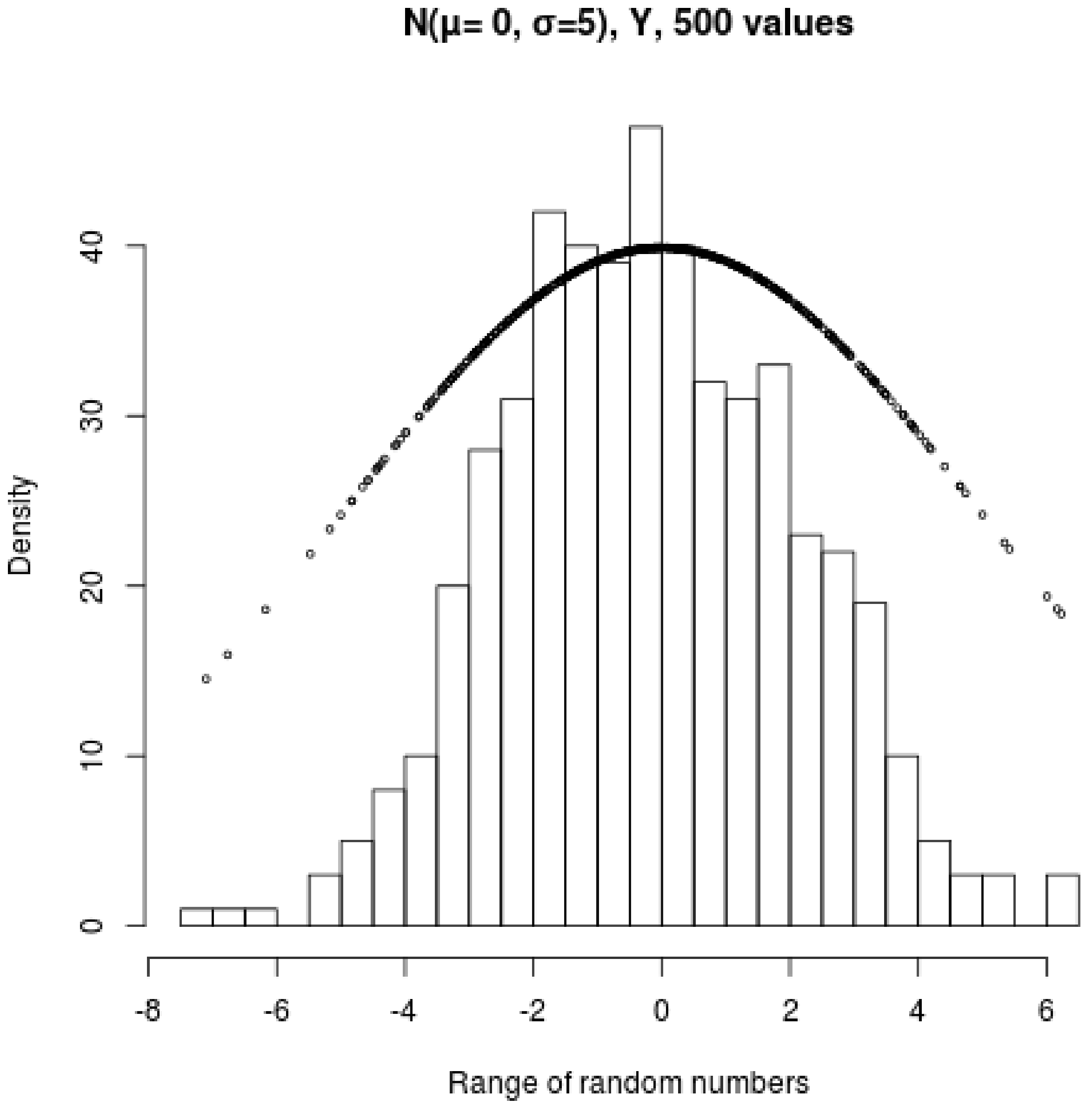

N(μ= 0, σ=5), X, 500 values
Density
0
10
20
30
40
50
-6
-4
-2
0
2
4
6
Range of random numbers


N(μ= 0, σ=5), Y, 500 values
Density
0
10
20
30
40
50
-6
-4
-2
0
2
4
6
Range of random numbers

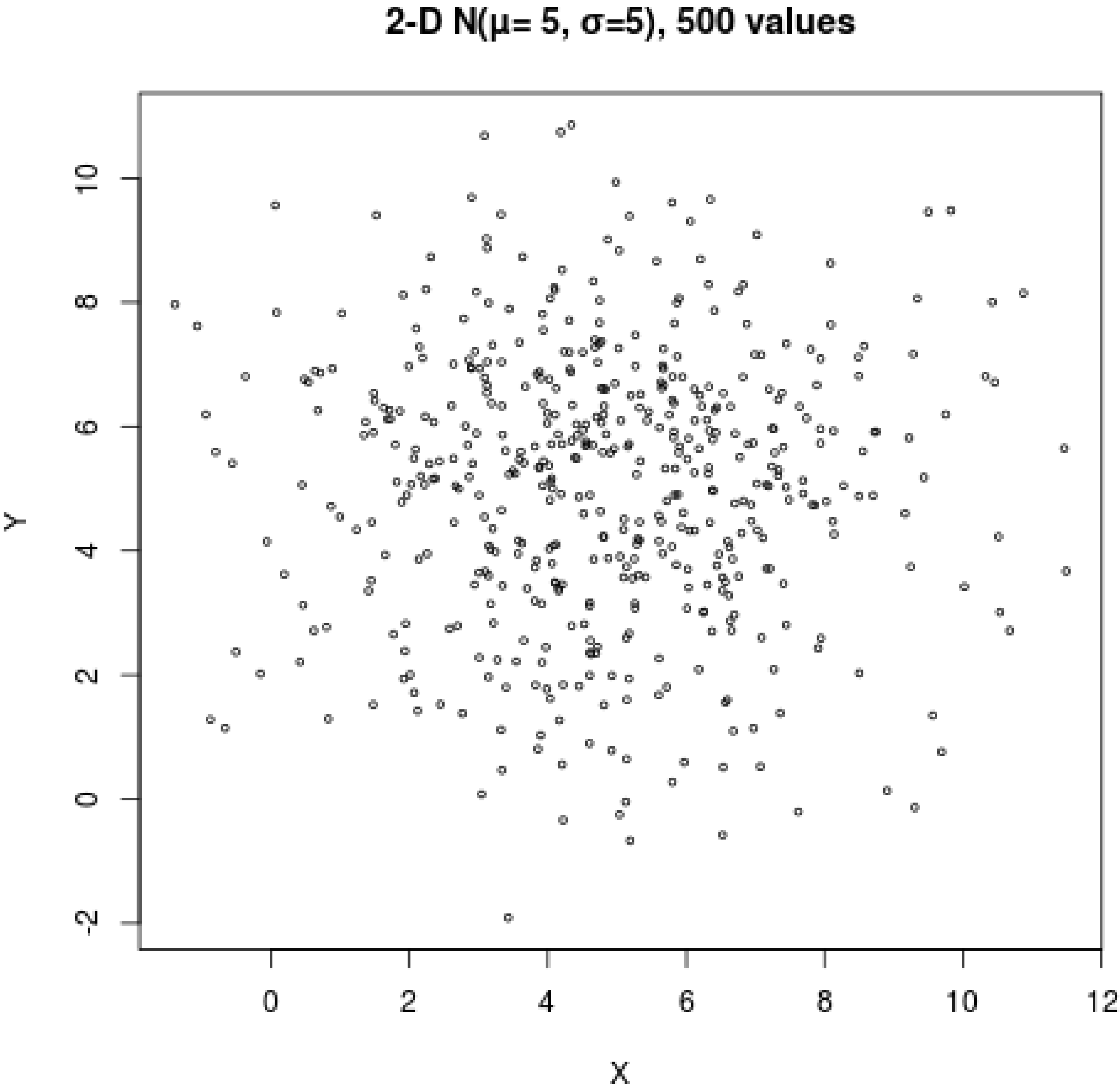


By Marsaglia-Bray

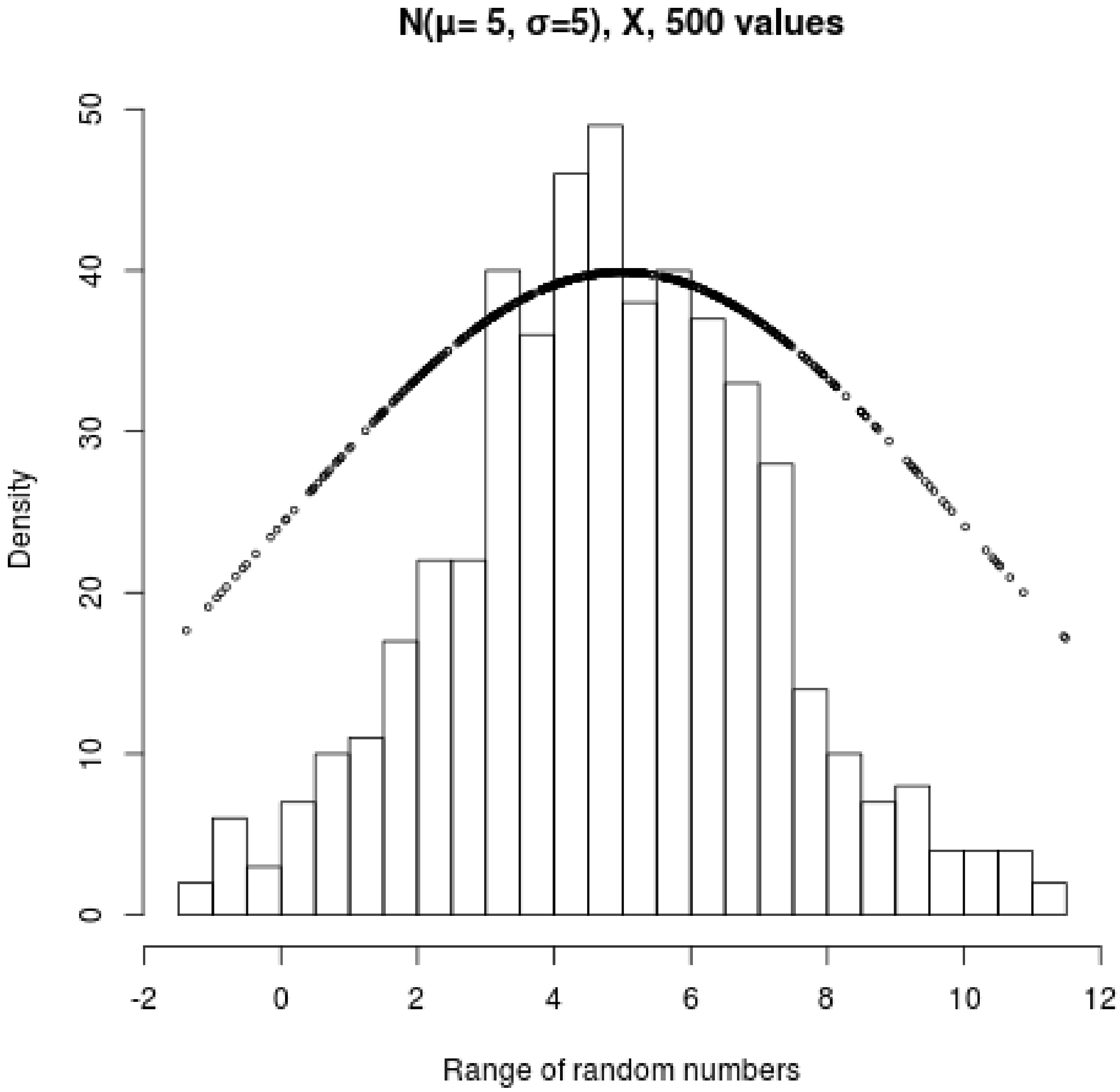

N(μ= 0, σ=5), X, 500 values
Density
0
20
40
60
80
-5
0
5
Range of random numbers

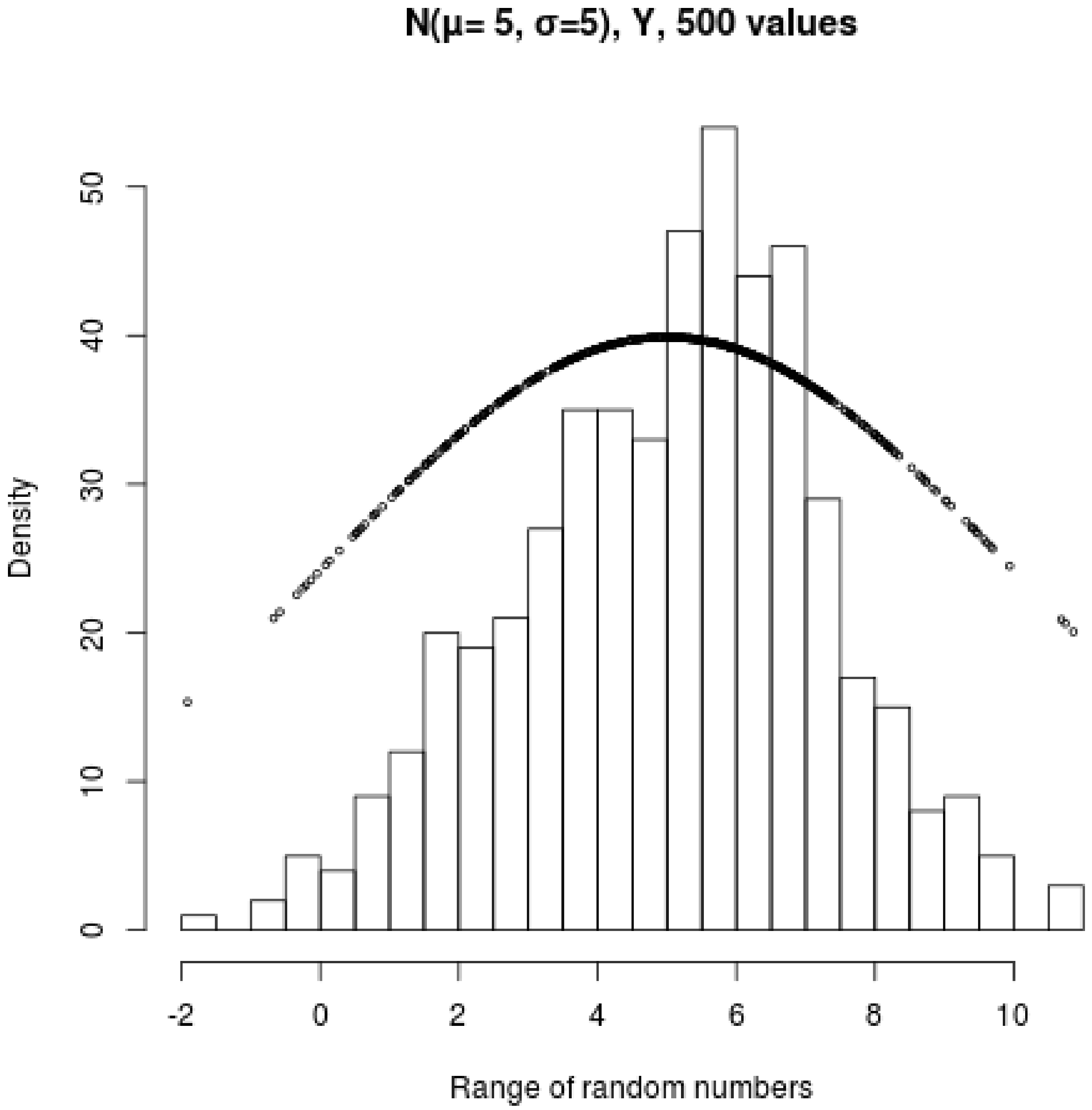

N(μ= 0, σ=5), Y, 500 values
Density
0
10
20
30
40
-8
-6
-4
-2
0
2
4
6
Range of random numbers

$N(\mu = 5, \sigma = 5)$ for 500 values

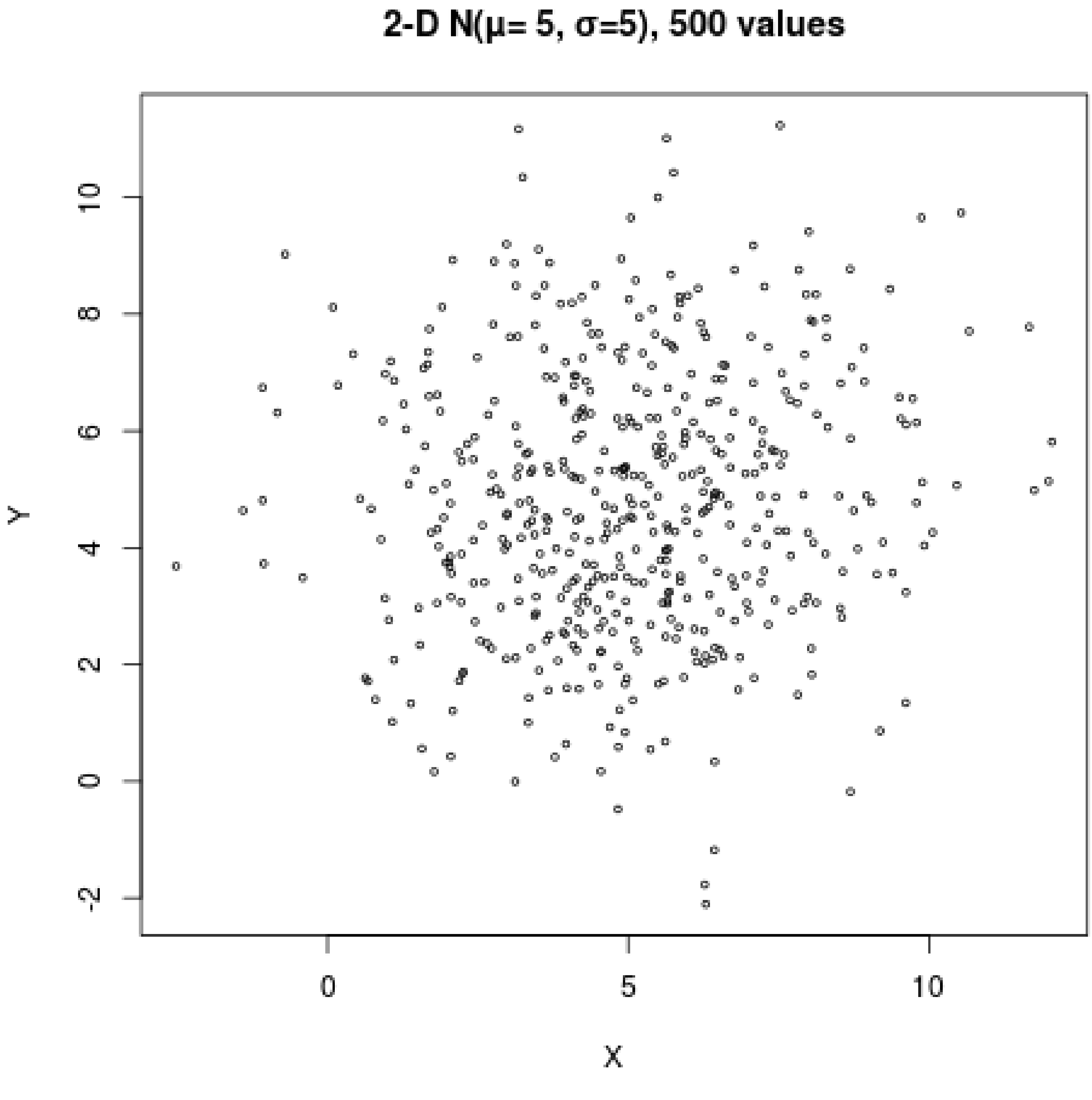


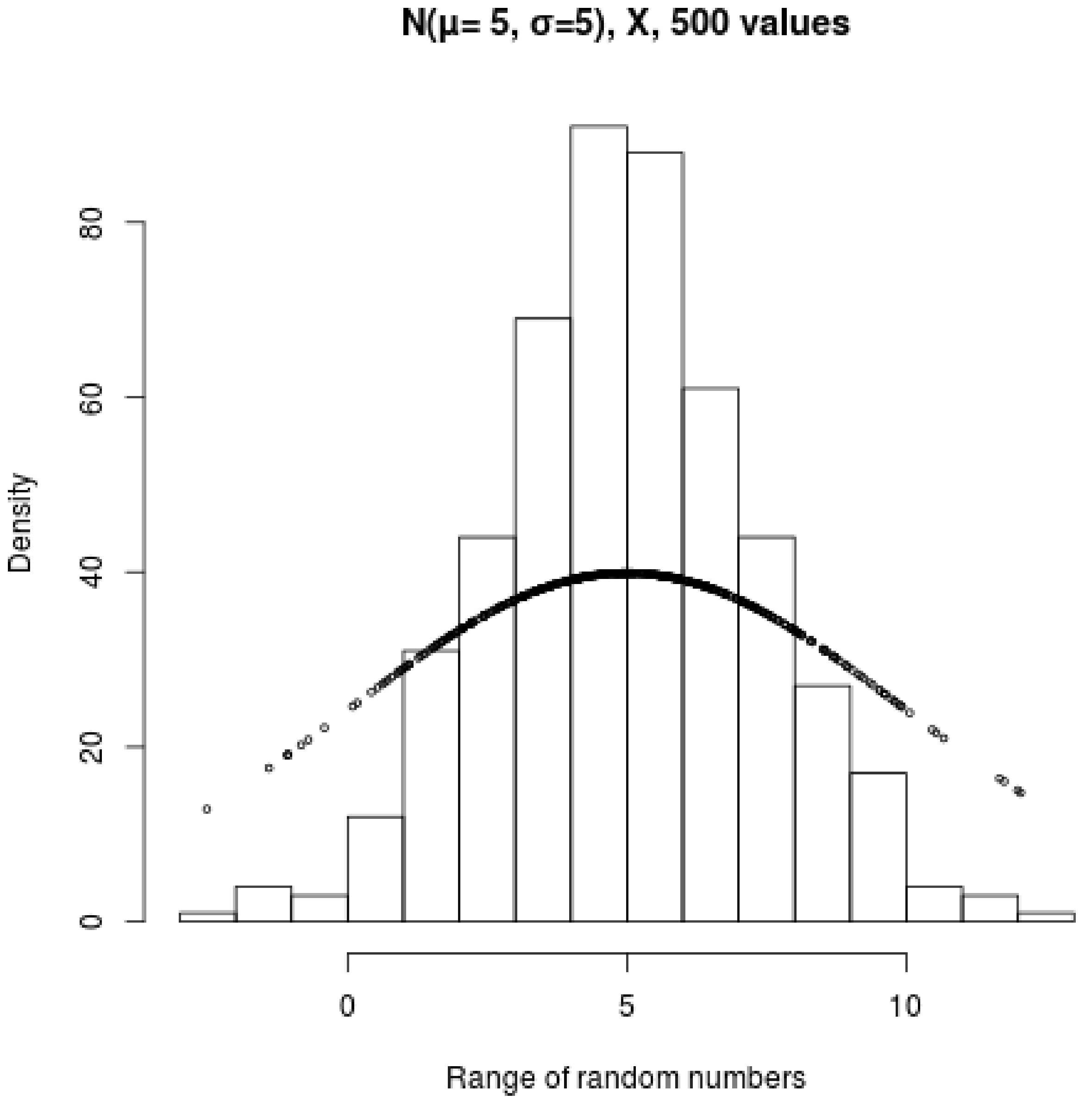
By Box-Muller

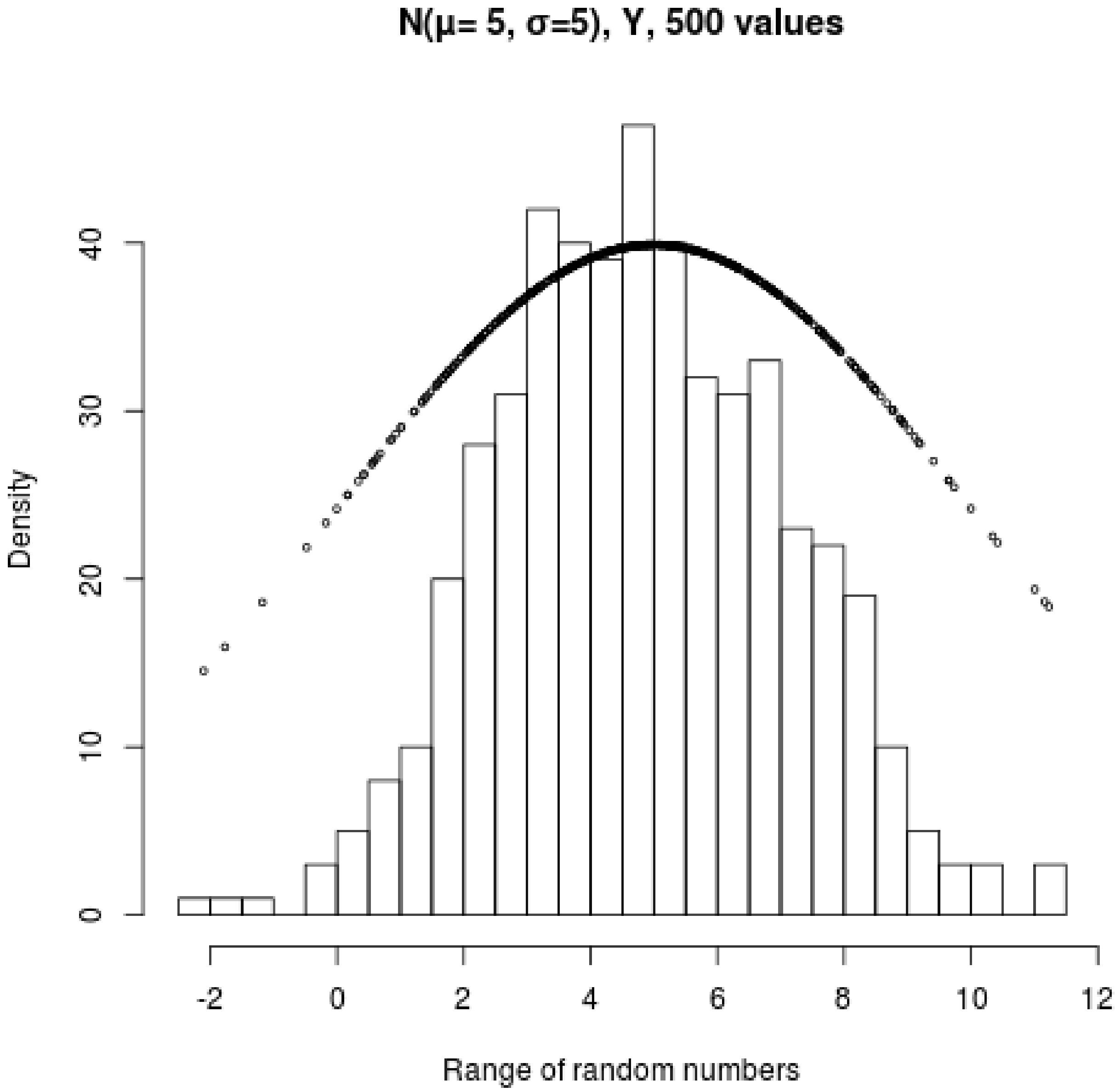

N(μ= 5, σ=5), X, 500 values
Density
0
10
20
30
40
50
-2
0
2
4
6
8
10
12
Range of random numbers


N(μ= 5, σ=5), Y, 500 values
Density
0
10
20
30
40
50
-2
0
2
4
6
8
10
Range of random numbers

By Marsaglia-Bray




N(μ= 5, σ=5), X, 500 values
Density
0
20
40
60
80
0
5
10
Range of random numbers


N(μ= 5, σ=5), Y, 500 values
Density
0
10
20
30
40
-2
0
2
4
6
8
10
12
Range of random numbers

5 Question 3

Box-Muller Method

Time taken for 10000 numbers generated = 0.01352119

Marsaglia-Bray Method

Time taken for 10000 numbers generated = 0.009799719

So, we se that Marsaglia-Bray Method is faster than Box-Muller Method.

6 Question 4

In Marsaglia-Bray Method

Rejection ratio, Theoretical = 0.2146018 , Generated = 0.2107963

Simulated rejection ratio is very close to theoretical one.