Lab3

Cui Qingxuan, Nisal Amashan

2025-02-09

Contents

1 Collaborations

Nisal Amashan: Responsible for the question 1.

Cui Qingxuan: Responsible for the question 2.

2 Question 1

3 Question 2

3.1 Generate a Random Vector using the Box-Muller Method

Measure the time for generating 10 000 000:

```
##    user  system elapsed
##    0.03    0.00    0.03
```

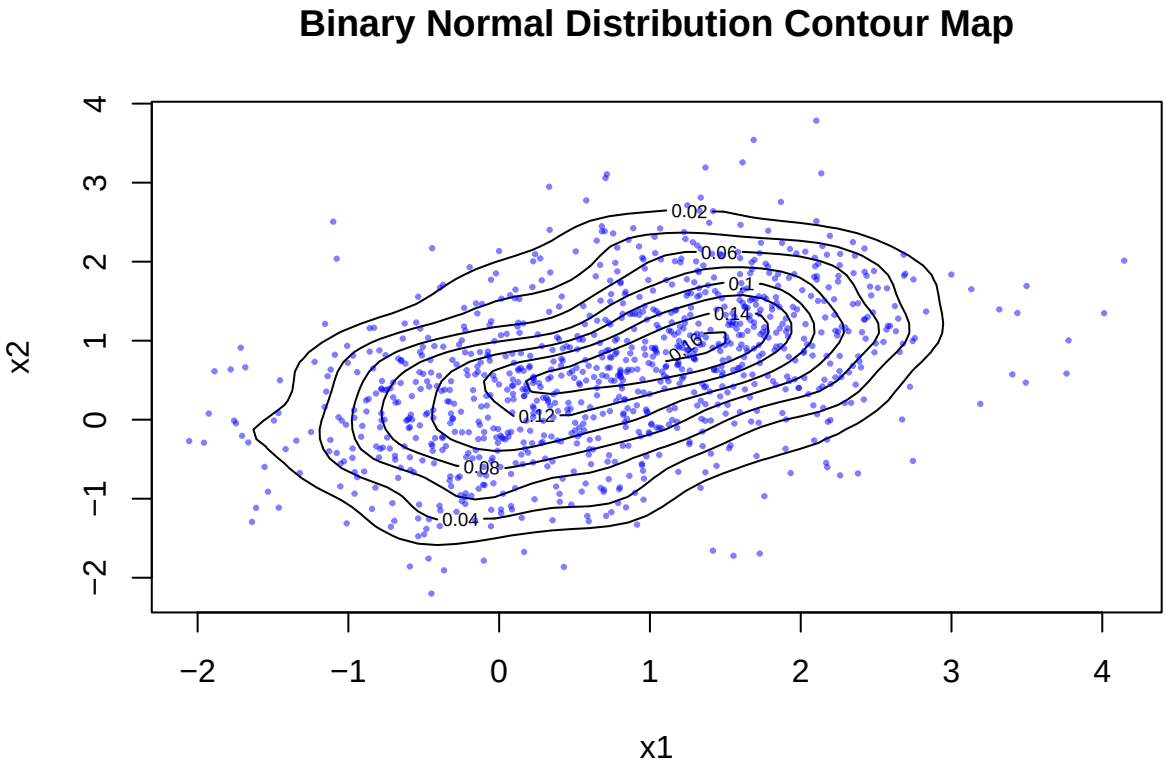
3.2 Generate a Random Vector using the Package mvtnorm

Measure the time for generating 10 000 000:

```
##    user  system elapsed
##       0       0       0
```

3.3 Generate the Random Vectors from 2 Distributions

Plot using 1000 Samples



4 Appendix

```
normRVgen = function(n, mu, sigma){
  # create 2 rv vector uniformly distribute
  # u1 represents R
  # u2 represents angle
  u1 = runif(n, min=0, max=1)
  u2 = 2*pi*runif(n, min=0, max=1)

  # Generate X=[x1, x2] ~ N(0,1)
  x1 = sqrt(-2*log(u1))*cos(u2)
  x2 = sqrt(-2*log(u1))*sin(u2)
  x = rbind(x1, x2)
  # Transform to Z via Z = A.T @ X + mu
  # Cholesky Decomposition -> get A from Sigma = A.T@A
  At = t(chol(sigma))
  Z = At %*% x + mu

  return (Z)
}

n =  1000
mu = c(0,0)
```

```r
sigma = matrix(c(0.6,0,0,0.6), nrow=2)
start = proc.time()
rv = normRVgen(n,mu,sigma)
end = proc.time()
print(end - start)

library(mvtnorm)
start = proc.time()
rv_buildin <- rmvnorm(n, mean = mu, sigma = sigma)
end = proc.time()
print(end-start)

# Generate 500 values for each distribution
n = 500
rv_d1 = normRVgen(n, mu, sigma)

mu_d2 = c(1.5, 1.2)
sigma_d2 = matrix(c(0.5,0,0,0.5), nrow=2)
rv_d2 = normRVgen(n, mu_d2, sigma_d2)
# Bind them and shuffle
rv_mix = as.data.frame(t(cbind(rv_d1, rv_d2)))
set.seed(123)
rv_shuffled = rv_mix[sample(nrow(rv_mix)), ]
colnames(rv_shuffled) = c("x1", "x2")

# Generate 500 values for each distribution
n = 500
rv_d1 = normRVgen(n, mu, sigma)

mu_d2 = c(1.5, 1.2)
sigma_d2 = matrix(c(0.5,0,0,0.5), nrow=2)
rv_d2 = normRVgen(n, mu_d2, sigma_d2)
# Bind them and shuffle
rv_mix = as.data.frame(t(cbind(rv_d1, rv_d2)))
set.seed(123)
rv_shuffled = rv_mix[sample(nrow(rv_mix)), ]
colnames(rv_shuffled) = c("x1", "x2")

# Plot the Function
library(MASS)
z <- kde2d(rv_shuffled$x1, rv_shuffled$x2, n = 50)
contour(z, xlab = "x1", ylab = "x2", main = "Binary Normal Distribution Contour Map")
points(rv_shuffled$x1, rv_shuffled$x2, col = rgb(0, 0, 1, 0.5), pch = 19,cex = 0.3)
```