

8)b) e 8)c)

```
mu <- c(0, 4)
Sigma <- matrix(c(1, 2, 2, 9), nrow = 2)

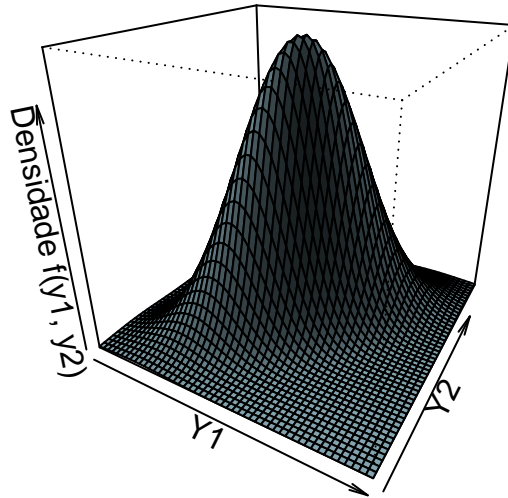
dbivnorm <- function(y1, y2, mu, Sigma) {
  y <- c(y1, y2)
  det_Sigma <- det(Sigma)
  inv_Sigma <- solve(Sigma)
  mahalanobis_sq <- t(y - mu) %*% inv_Sigma %*% (y - mu)
  densidade <- (1 / (2 * pi * sqrt(det_Sigma))) * exp(-0.5 * mahalanobis_sq)
  return(densidade)}

y1_seq <- seq(mu[1] - 3 * sqrt(Sigma[1,1]), mu[1] + 3 * sqrt(Sigma[1,1]), length.out = 50)
y2_seq <- seq(mu[2] - 3 * sqrt(Sigma[2,2]), mu[2] + 3 * sqrt(Sigma[2,2]), length.out = 50)
grid_pontos <- expand.grid(Y1 = y1_seq, Y2 = y2_seq)

densidade_valores <- mapply(dbivnorm,
  y1 = grid_pontos$Y1,
  y2 = grid_pontos$Y2,
  MoreArgs = list(mu = mu, Sigma = Sigma))
densidade_grid <- matrix(densidade_valores, nrow = length(y1_seq), ncol = length(y2_seq))

persp(x = y1_seq, y = y2_seq, z = densidade_grid,
  main = "Gráfico da Função de Densidade Normal Bivariada",
  xlab = "Y1", ylab = "Y2", zlab = "Densidade f(y1, y2)",
  theta = 30, phi = 25, col = "lightblue", shade = 0.4)
```

Gráfico da Função de Densidade Normal Bivariada



9d)

```
num <- 1000000
lancamentos <- 12
resultado_desejado <- c(2, 2, 2, 2, 2, 2)
sucessos <- 0

for (i in 1:num) {
  resultado_iteracao <- sample(1:6, size = lancamentos, replace = TRUE)
  contagem_faces <- table(factor(resultado_iteracao, levels = 1:6))
  if (all(contagem_faces == resultado_desejado)) {
    sucessos <- sucessos + 1}}

probabilidade_aproximada <- sucessos / num

print(paste("Número de simulações:", num))

## [1] "Número de simulações: 1e+06"

print(paste("Probabilidade aproximada:", probabilidade_aproximada))

## [1] "Probabilidade aproximada: 0.003434"
```