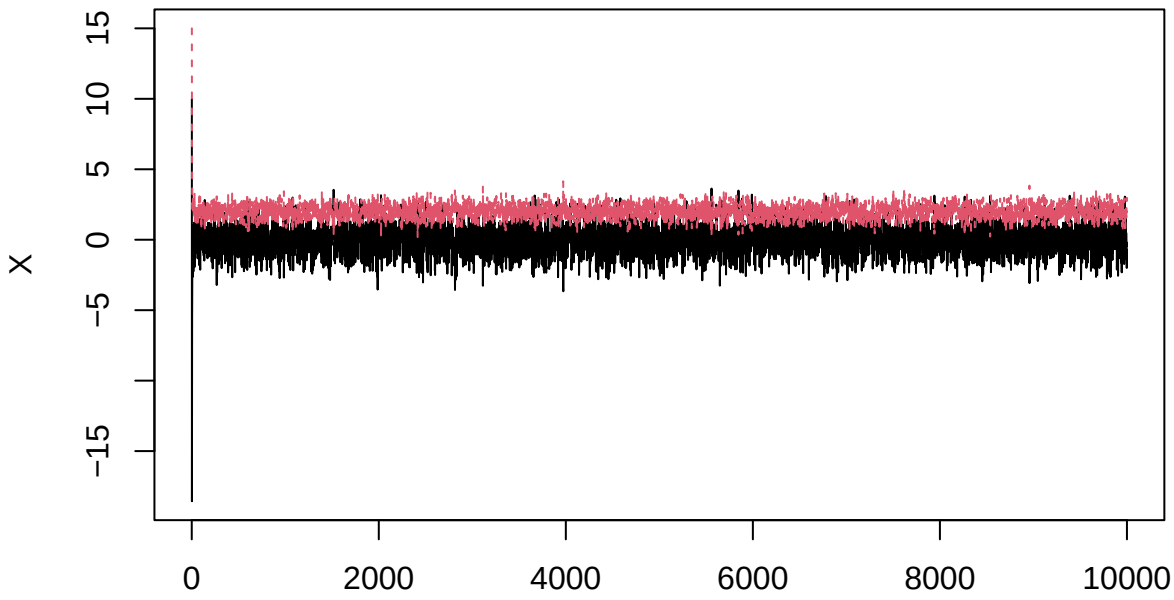
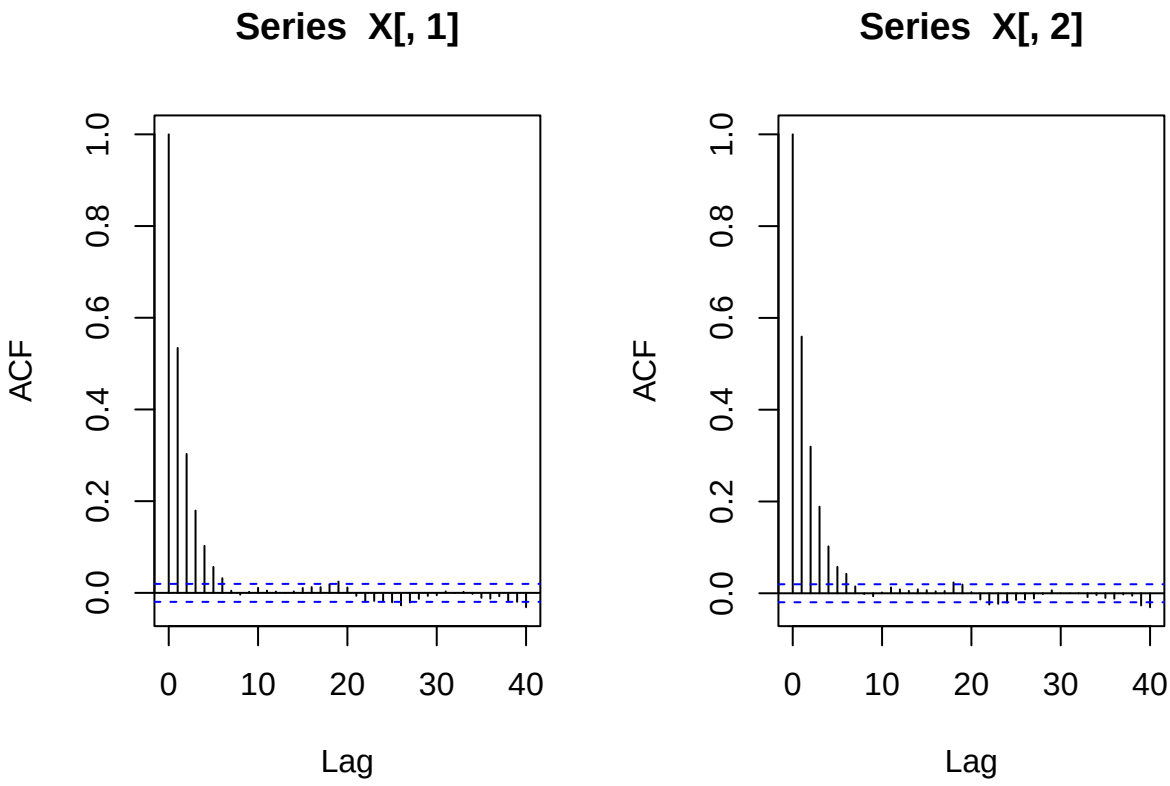
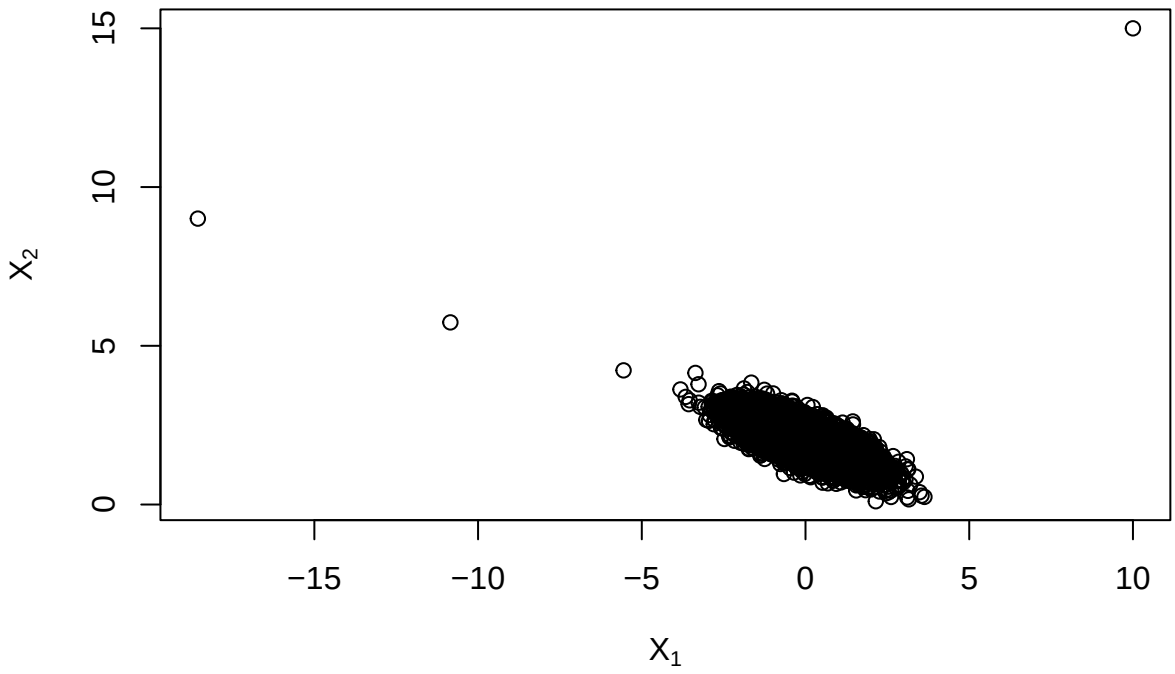
```r
## Definindo os parametros da Normal bivariada.
rho <- -.75
mu1 <- 0
mu2 <- 2
sigma1 <- 1
sigma2 <- .5
s1 <- sqrt(1 - rho^2) * sigma1
s2 <- sqrt(1 - rho^2) * sigma2

## Definindo a quantidade de interações a serem realizadas.
N <- 10000

## Matriz para armazenar as amostras.
X <- matrix(0, N, 2)

## Valores iniciais, seria interessante deixar eles aleatórios.
X[1, ] <- c(10, 15)

## Gerando a cadeia.
for (i in 2:N) {
  x2 <- X[i-1, 2]
  m1 <- mu1 + rho * (x2 - mu2) * sigma1/sigma2
  X[i, 1] <- rnorm(1, m1, s1)
  x1 <- X[i, 1]
  m2 <- mu2 + rho * (x1 - mu1) * sigma2/sigma1
  X[i, 2] <- rnorm(1, m2, s2)
}

## Plotagem das cadeis.
matplot(X, type = "l")
```




```
## Correlacao entre os valores.
par(mfrow = c(1, 2))
acf(X[,1])
acf(X[,2])
```



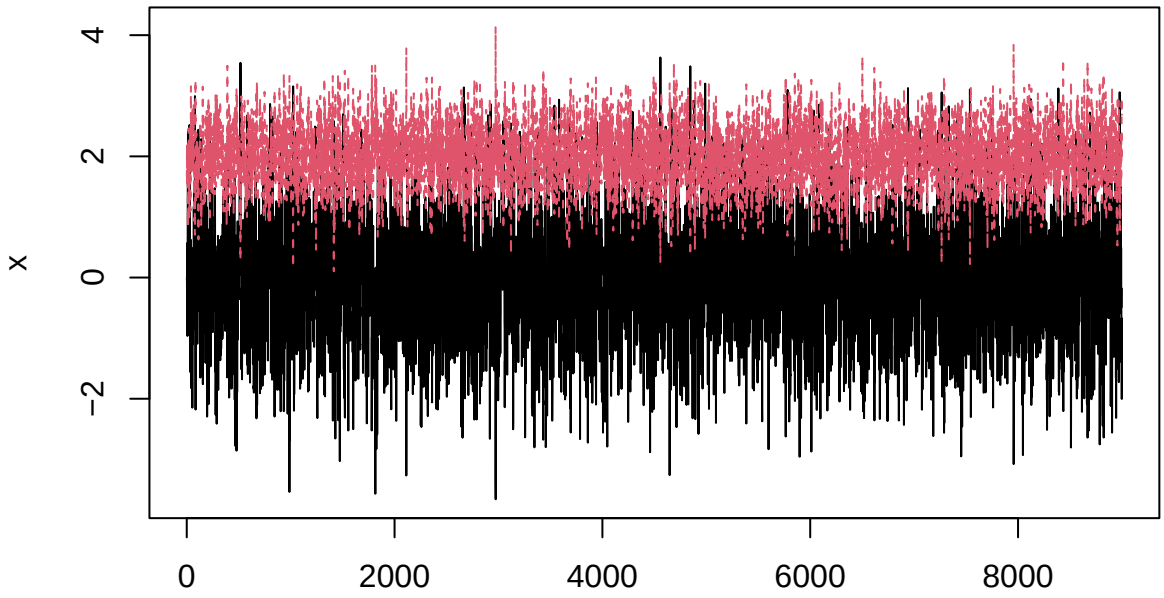

```
## Conjunta.
par(mfrow = c(1, 1))
plot(X, main = "", xlab = bquote(X[1]),
     ylab = bquote(X[2]), ylim = range(X[, 2]))
```



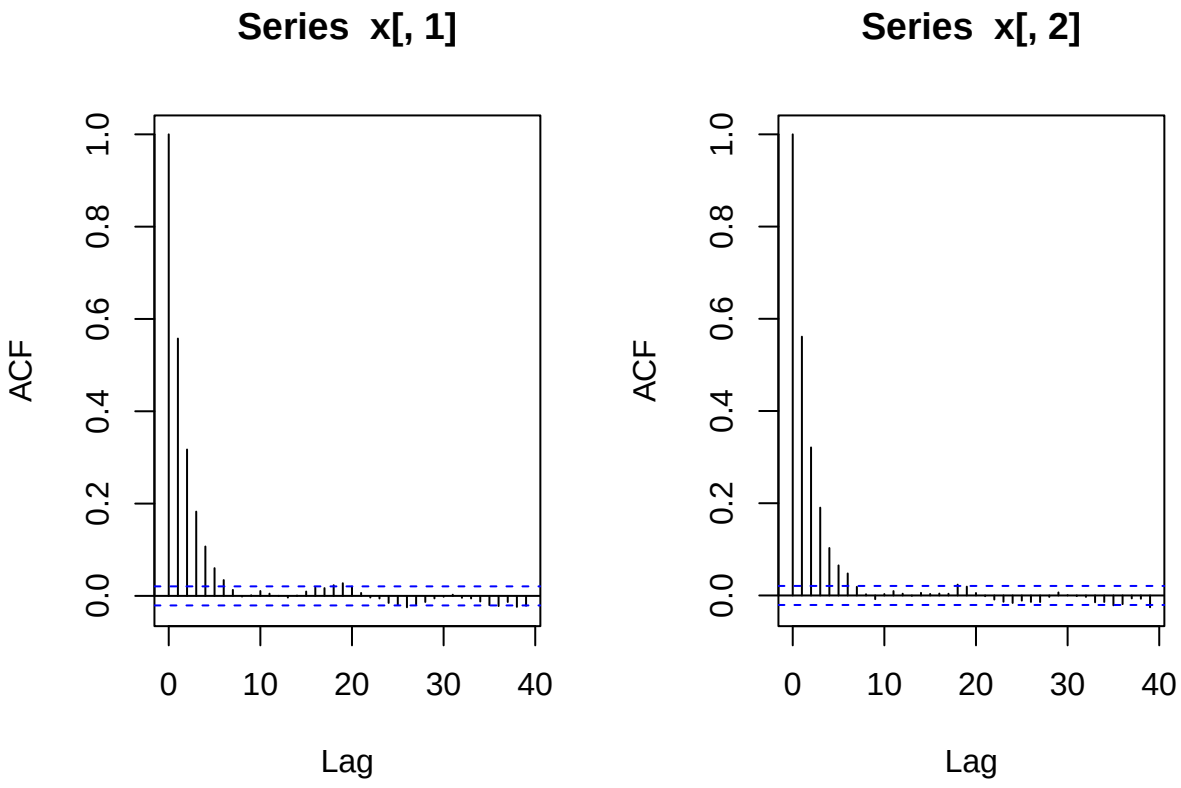

```
## Definindo a quantidade de valores que serão descartados do começo.
burn <- 1000

## Descartando-os.
b <- burn + 1
x <- X[b:N, ]

## Tamanho da amostra após o burn
dim(x)
```

```
## [1] 9000    2
```

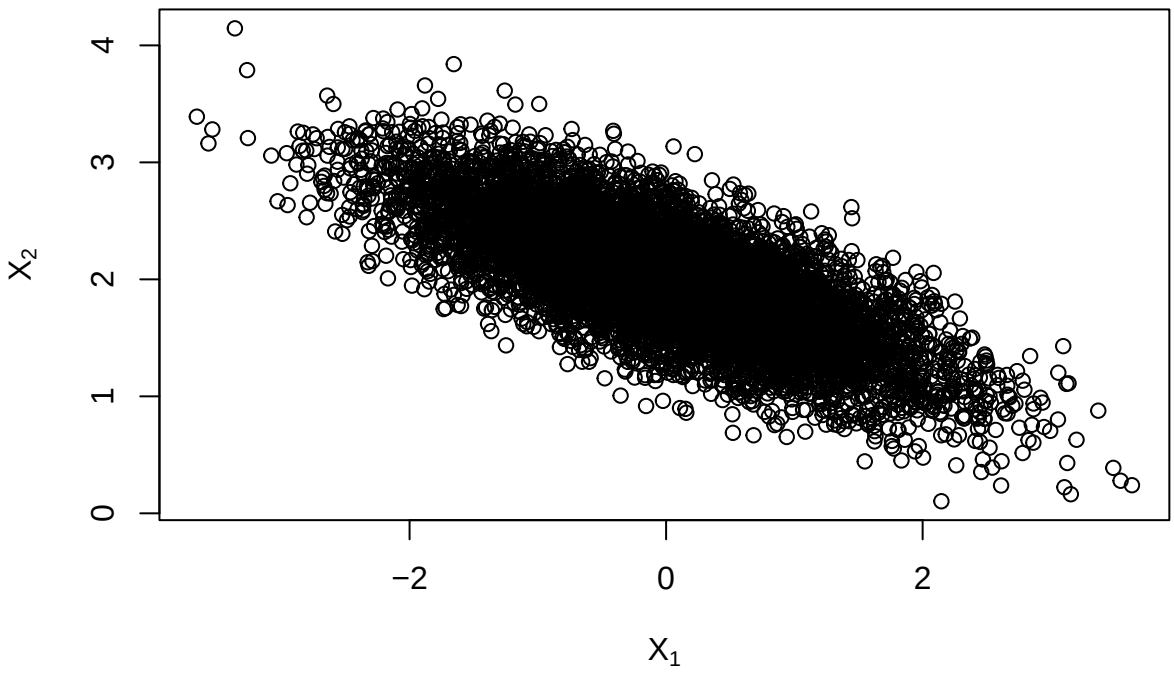
```
## Plotagem do gráfico
matplot(x, type = "l")
```




```
## Nao elimina o problema de autocorrelacao.
par(mfrow = c(1, 2))
acf(x[,1])
acf(x[,2])
```



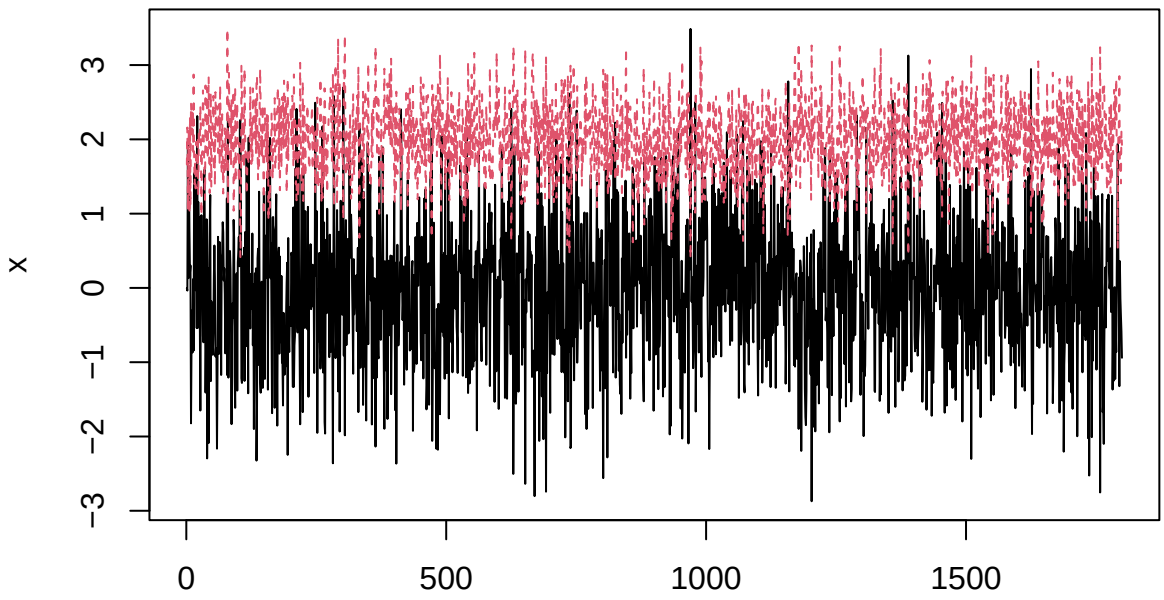

```
## Mas elimina o problema dos valores iniciais discrepantes
par(mfrow = c(1, 1))
plot(x, main = "", xlab = bquote(X[1]),
     ylab = bquote(X[2]), ylim = range(x[, 2]))
```



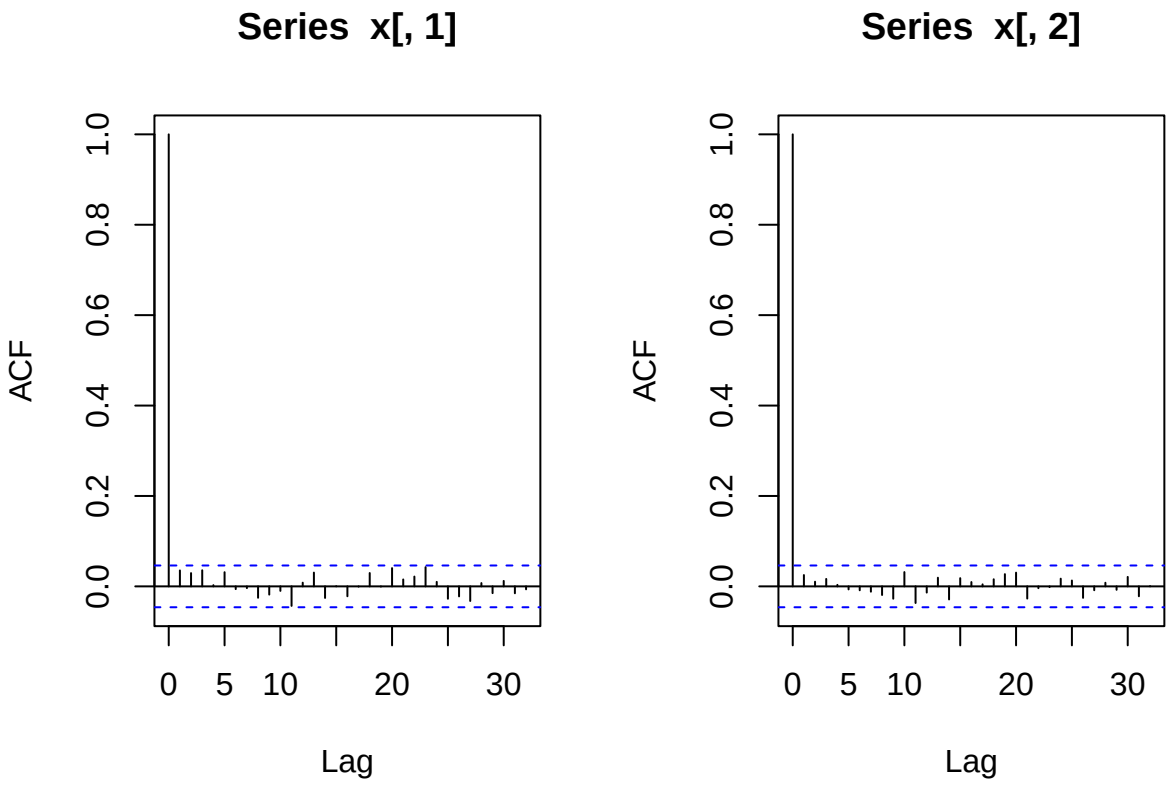

```
## Fazendo o thinning, método para eliminar a autocorrelação.
x <- x[seq(1, nrow(x), 5), ]
dim(x)
```

```
## [1] 1800    2
```

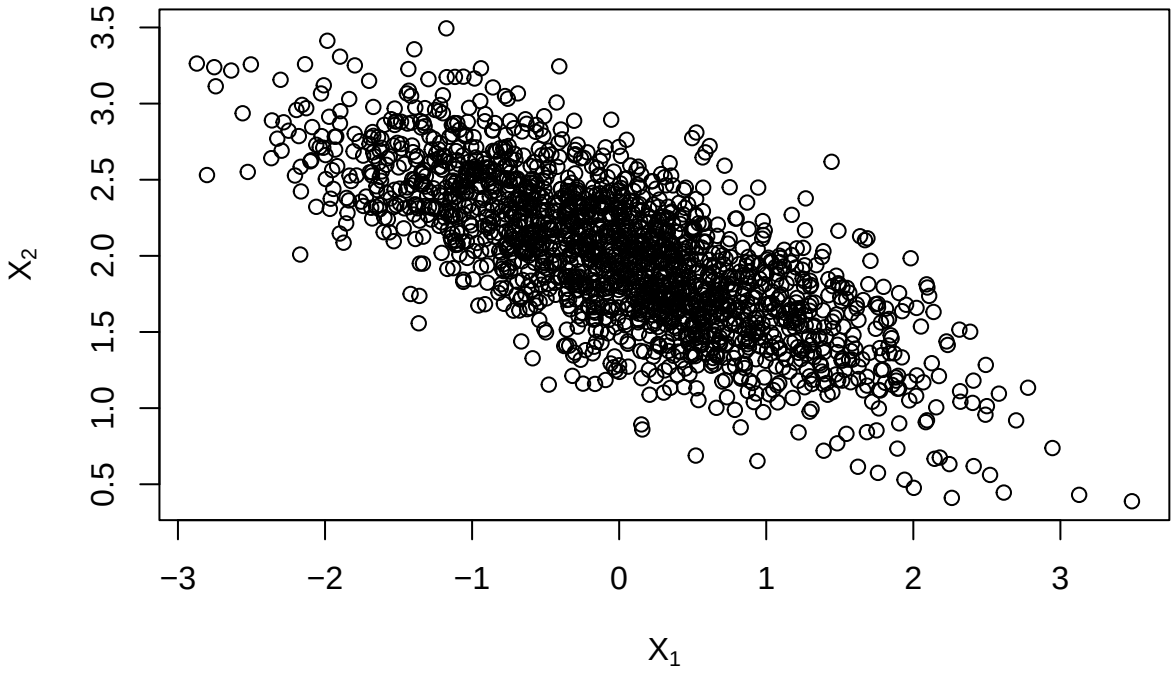
```
## Confere novamente pelo gráfico
matplot(x, type = "l")
```



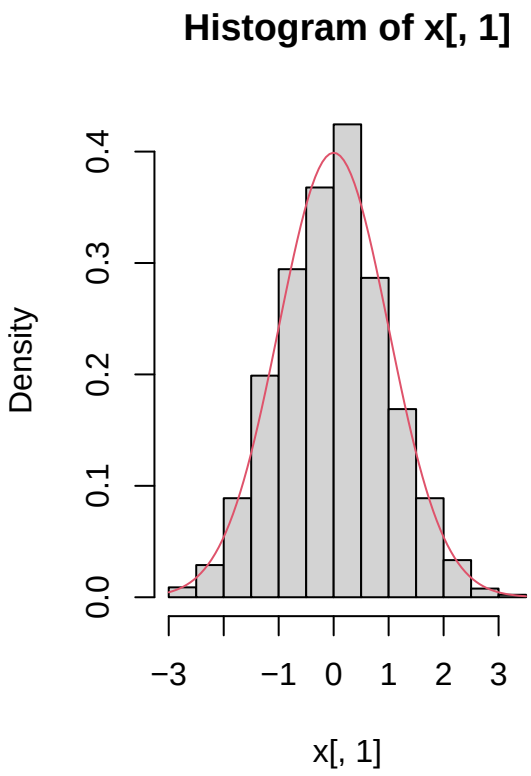

```
## Agora elimina o problema de autocorrelacao
par(mfrow = c(1, 2))
acf(x[,1])
acf(x[,2])
```

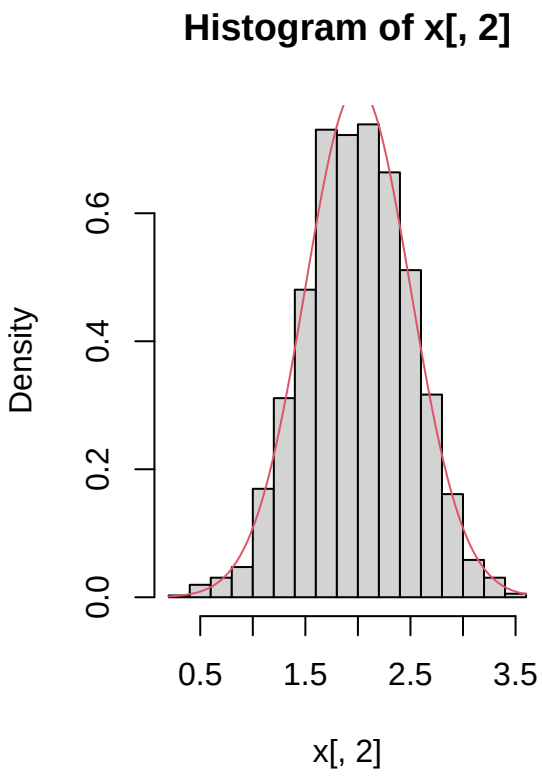



```
## Conjunta
par(mfrow = c(1, 1))
plot(x, main = "", xlab = bquote(X[1]),
     ylab = bquote(X[2]), ylim = range(x[, 2]))
```




```r
## Compara com as marginais
# No caso da normal bivariada, as marginais são normais com seus respectivos parâmetros.
par(mfrow = c(1, 2))
hist(x[, 1], freq = FALSE)
curve(dnorm(x, mu1, sigma1), col = 2, add = TRUE)
hist(x[, 2], freq = FALSE)
curve(dnorm(x, mu2, sigma2), col = 2, add = TRUE)
```






```
par(mfrow = c(1, 1))

## Comparando as estatisticas
colMeans(x)
```

```
## [1] -0.002016516  1.996076105
```

```
cov(x)
```

```
##            [,1]       [,2]
## [1,]  0.9522941 -0.3607791
## [2,] -0.3607791  0.2482533
```

```
cor(x)
```

```
##            [,1]       [,2]
## [1,]  1.0000000 -0.7420076
## [2,] -0.7420076  1.0000000
```