

# Generación de arreglos pseudoaleatorios en MATLAB

**Objetivos.** Aprender los comandos MATLAB para construir vectores y matrices pseudoaleatorios.

**Requisitos.** Tener instalado el sistema MATLAB o alguno de sus análogos libres: GNU Octave, Scilab, FreeMat.

**1. Números pseudoaleatorios reales uniformemente distribuidos en el intervalo (0, 1).** Se recomienda ejecutar los siguientes comandos en el intérprete:

```
rand()
rand()
rand()
rand(3, 2)
rand(4, 1)
rand(1, 3)
rand(3)
```

**2. Análisis estadístico elemental de un vector pseudoaleatorio.**

```
n = 10;
a = rand(n, 1)
plot(a);
plot(sort(a))
length(a)
sum(a)
sum(a) / length(a)
mean(a)
sum((a - mean(a)) .^ 2) / (length(a) - 1)
var(a)
a <= 4
sum(a <= 4)
```

Luego se recomienda repetir el mismo análisis para un vector más largo, es decir, repetir los comandos anteriores, pero sustituir  $n = 10$  por  $n = 10000$ .

### 3. Vector real con componentes pseudoaleatorias uniformemente distribuidas en el intervalo (5, 7).

```
ones(16, 1)
v = 5 * ones(16, 1) + 2 * rand(16, 1)
plot(v)
min(v)
max(v)
mean(v)
sum(v <= 6)
M = 5 * ones(2, 3) + 2 * rand(2, 3)
```

### 4. Vector de componentes reales pseudoaleatorias uniformemente distribuidas en el intervalo (-3, 1).

```
a = -3; b = 1; n = 100000;
v = a * ones(n, 1) + (b - a) * rand(n, 1);
[min(v), max(v), mean(v)]
sum(v <= -1)
sum(v <= -1) / n
(-1 - a) / (b - a)
hist(v, 10);
hist(v, 20);
```

En versiones modernas de MATLAB se recomienda usar la función `histogram` en vez de `hist`. En GNU Octave no hay función `histogram`, solamente hay `hist`.

### 5. Ejercicios.

- Generar una matriz  $5 \times 3$  de números reales pseudoaleatorios uniformemente distribuidos en el intervalo  $(-1, 1)$ .
- Generar un vector de longitud 20 de números reales pseudoaleatorios uniformemente distribuidos en el intervalo  $(-2, 2)$ . Mostrar el valor promedio, el valor mínimo y el valor máximo de las componentes del vector.
- Generar un vector de longitud 10000 de números reales pseudoaleatorios uniformemente distribuidos en el intervalo  $(-2, 2)$ . Calcular la cantidad de las componentes del vector que son menores o iguales a 1. Calcular la porción de las componentes del vector que son menores o iguales a 1. Dividir la longitud del intervalo  $(-2, 1)$  entre la longitud del intervalo  $(-2, 2)$ .

### 6. Números pseudoaleatorios enteros.

```
randi([-2, 2], 10)
randi([-2, 2], 5, 3)
randi([-2, 2], 6, 1)
v = randi([-2, 2], 10000, 1);
hist(v);
```

**7. Números pseudoaleatorios reales con distribución de Gauss.** La distribución de Gauss se conoce también como la distribución normal.

```
randn(5)
randn(3, 4)
v = randn(10, 1)
[ min(v), max(v), mean(v) ]
v = randn(10, 1)
[ min(v), max(v), mean(v) ]
v = randn(10000, 1);
[ min(v), max(v), mean(v) ]
hist(v, 20)
var(v)
```

**8. Números pseudoaleatorios reales con distribución de Gauss, modificando la media y la varianza.**

```
n = 100000;
v = -3 * ones(n, 1) + 10 * randn(n, 1);
[ mean(v), var(v) ]
hist(v);
plot(sort(v));
```

## 9. Ejercicios.

- Generar una matriz pseudoaleatoria  $5 \times 3$  con entradas normalmente distribuidas.
- Generar un vector pseudoaleatorio de longitud  $10^4$  con componentes normalmente distribuidas, con media 5 y varianza 4. Hacer un análisis estadístico elemental, aplicando los comandos `mean`, `var` y `hist`.

## 10. Permutaciones pseudoaleatorias.

```
randperm(6)
randperm(6)
randperm(6)
randperm(10)
```