

Criando Algoritmo em R para Geração de Matrizes

Neste breve post vou desenvolver um algoritmo para criação de matrizes que será usado em outros posts envolvendo Álgebra Linear. O algoritmo vai solicitar como input as seguintes variáveis:

- Qtd de Linhas i
- Qtd de Colunas j
- Média dos valores x
- Desvio padrão dos valores u

O algoritmo poderá gerar matrizes do tipo Quadrada ($i = j$) ou retangular ($i \neq j$). Primeiro vou montar o algoritmo com um passo a passo e depois empacotar ele em uma função com o nome de **gen_matrix(i,j,x,u)**.

```
# Criando as variáveis do algoritmo
i <- 5 # Linhas da matriz
j <- 5 # Colunas
x <- 10 # Média dos elementos
u <- 2 # Desvio padrão

# Criando variável com números aleatórios da matriz
rdm <- rnorm(n = i*j,
             mean = x,
             sd = u)

# Arredondando os valores
rdm_adj <- round(rdm, digits = 0)

# Criando uma variável com a matriz
mt <- matrix(data = rdm_adj,
             nrow = i,
             ncol = j)

print(mt)
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    7   10    8   11   11
## [2,]   10    9   14    7   10
## [3,]   11    9   12   10    6
## [4,]    9   11    5    7   12
## [5,]   14   10   12    7    8
```

Funcionou muito bem. Agora podemos empacotar estes passos em uma função para executá-la quando necessário.

```
gen_matrix <- function(i,j,x,u){
  mt <- matrix(data = round(rnorm(n = i*j,
                                  mean = x,
                                  sd = u),
                                  digits = 0),
               nrow = i,
               ncol = j)
  print(mt)
}

gen_matrix(4,4,10,2)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]  11   9  11  12
## [2,]   7   9  10  10
## [3,]   9   7   7   8
## [4,]   8   7   8  11
```

Podemos aprimorar esta função e adicionar nela a função **pander()** no lugar de **print()**, que irá aplicar um design mais amigável no output final. Desta forma a função ficará:

```
gen_matrix <- function(i,j,x,u){
  mt <- matrix(data = round(rnorm(n = i*j,
                                  mean = x,
                                  sd = u),
                                  digits = 0),
               nrow = i,
               ncol = j)
  pander::pander(mt)
}

gen_matrix(3,3,5,2)
```

5	4	5
8	8	6
7	8	5

```
gen_matrix(6,6,50,25)
```

48	22	67	91	45	13
34	25	40	7	12	52
12	80	76	52	41	10
26	71	61	96	27	68
82	19	94	71	30	80
18	5	83	90	76	22
