ideia

April 5, 2025

1 Sequências aleatórias

[4]:
```
import pandas as pd
import numpy as np
```

1.1 Função de Matriz Principal

Tomaremos um array de tamanho n, no qual n seja divisivel por 60.

[5]:
```
def matriz(num_colunas, array1):
    """
    Gera uma matriz sequencial a partir de um array, com o número de colunas␣
 ↪especificado.

    Args:
        array (list ou np.ndarray): Array de entrada.
        num_colunas (int): Número de colunas desejado na matriz.

    Returns:
        np.ndarray: Matriz sequencial.
    """
    if num_colunas > len(array1):
        raise ValueError("O número de colunas não pode ser maior que o tamanho␣
 ↪do array.")

    # Número de linhas na matriz
    num_linhas = len(array1) - num_colunas + 1

    # Criando a matriz sequencial
    matriz = np.array([array1[i:i + num_colunas] for i in range(num_linhas)])
    return matriz
```

1.1.1 Teste 1

- Cria-se um array com entrada n = 420
- Transforma-se esse array em uma matriz com tamanho de colunas fixo num_colunas = 60
- Espera-se retorno de uma matriz com tamanho 361x60

[11]:
```
array_teste = np.arange(1, 421)
print("Tamanho array de teste:", len(array_teste))
matriz_teste = matriz(60, array_teste)
print(f'Shape matriz de teste:{matriz_teste.shape}')
```

```
Tamanho array de teste: 420
Shape matriz de teste:(361, 60)
```

1.2 Carregando dados para testes

Aqui carrega-se uma data especifica de algum de dia de coleta e realize-se o ajuste necessários.

[12]:
```
data = pd.read_csv('/home/ozielramos/Documentos/Out/dados/Saidas/FUNCOES/DOUBLE␣
 ↪- 17_09_s1.csv')
data.head
```

[12]:
```
<bound method NDFrame.head of       22,11    0,5  0,5.1  1  BET Entrada  Odd
P60  P120  P180  …  \
0          0  22,11  -0,25  1    0    1,83    0  NaN   NaN   NaN  …
1          1  22,11     -1  0    0    1,07    0  NaN   NaN   NaN  …
2          2  22,11   1,75  2    0   24,83    1  NaN   NaN   NaN  …
3          3  22,11   1,75  2    0   25,25    1  NaN   NaN   NaN  …
4          4  22,11   1,75  2    0    8,55    1  NaN   NaN   NaN  …
…        …     …    … . .  …      …  …  …  …    …   …  …
2511    2511  20,11     -1  0    0       1    0  0.0   0.0   0.0  …
2512    2512  20,11     -1  0    0       1    0  0.0   0.0   0.0  …
2513    2513  20,11     -1  0    0       1    0  0.0   0.0   0.0  …
2514    2514  20,11     -1  0    0       1    0  0.0   0.0   0.0  …
2515    2515  20,11     -1  0    0       1    0  0.0   0.0   0.0  …

                P(1)          P(0)   LOG(P(1);2)     LOG(P(2);2)  Unnamed: 125  \
0                NaN           NaN           NaN             NaN           NaN
1                NaN           NaN           NaN             NaN           NaN
2                NaN           NaN           NaN             NaN           NaN
3                NaN           NaN           NaN             NaN           NaN
4                NaN           NaN           NaN             NaN           NaN
…                …             …             …               …             …
2511  0,04761904762  0,9523809524  -4,392317423  -0,07038932789           1.0
2512              0             1        #NUM!               0           0.0
2513  0,04761904762  0,9523809524  -4,392317423  -0,07038932789           1.0
2514  0,09523809524  0,9047619048  -3,392317423   -0,1443899093           2.0
2515  0,04761904762  0,9523809524  -4,392317423  -0,07038932789           1.0

      Unnamed: 126  Unnamed: 127  Unnamed: 128  Unnamed: 129  Unnamed: 130
0              NaN           NaN           NaN           NaN           NaN
1              NaN           NaN           NaN           NaN           NaN
2              NaN           NaN           NaN           NaN           NaN
3              NaN           NaN           NaN           NaN           NaN
```

```
4              NaN           NaN           NaN           NaN              NaN
…              …             …             …             …                …
2511          19.0          0,05          0,95  -4,321928095  -0,07400058144
2512          20.0             0             1         #NUM!                0
2513          19.0          0,05          0,95  -4,321928095  -0,07400058144
2514          18.0           0,1           0,9  -3,321928095   -0,1520030934
2515          19.0          0,05          0,95  -4,321928095  -0,07400058144

[2516 rows x 131 columns]>
```

[13]:
```
array_data = []
for i in range(len(data)):
    array_data.append(data['Entrada'][i])
print("Tamanho array de teste:", len(array_data))
```

```
Tamanho array de teste: 2516
```