

# NumPy

Introdução

Indexação booleana

## Indexação booleana

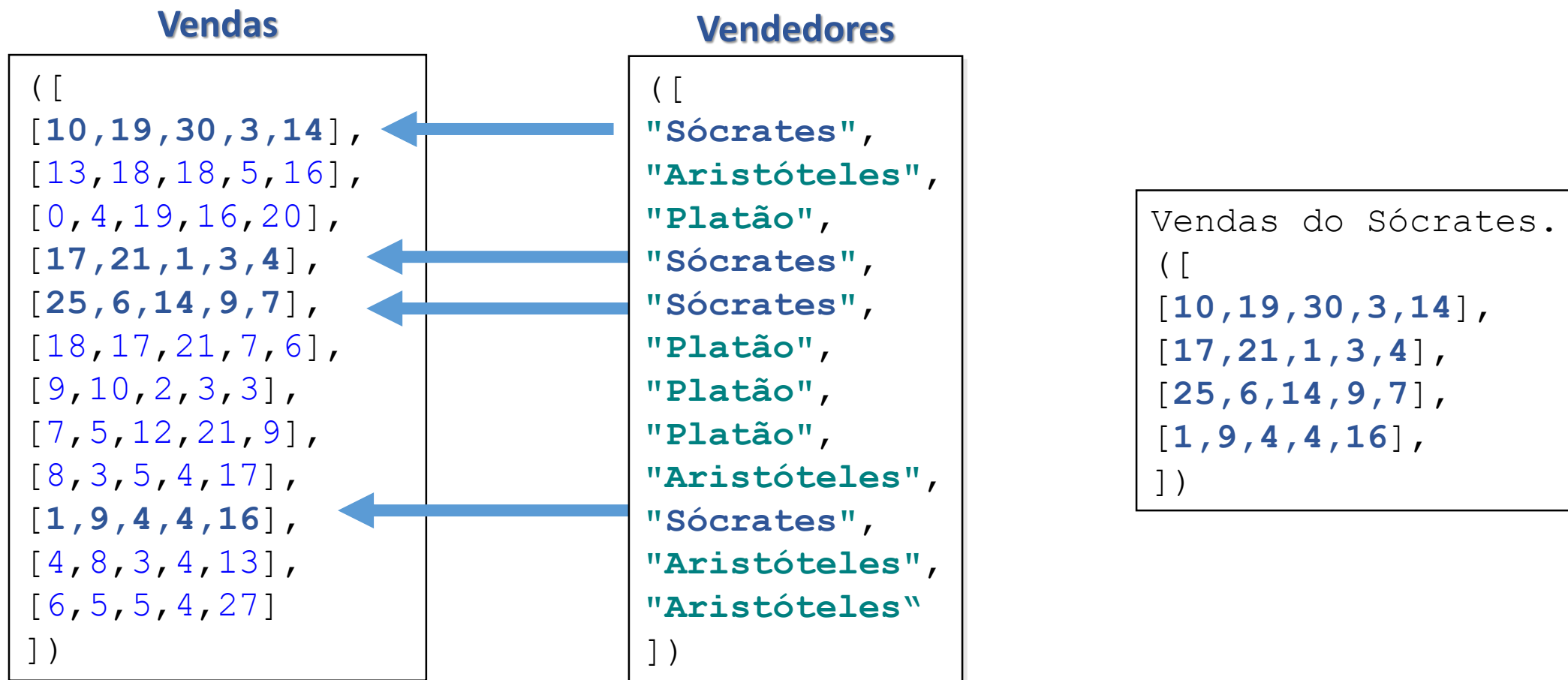
Nesta aula vou mostrar como podemos utilizar indexação booleana em arrays.

Vamos imaginar o seguinte cenário:

Temos um array com o número de vendas mensais de três vendedores, onde cada linha representa uma semana e cada coluna, um dia.

Temos também um array com os nomes dos vendedores na mesma ordem, significando que cada linha no array de vendas representa uma linha no array de vendedores.

## Indexação booleana



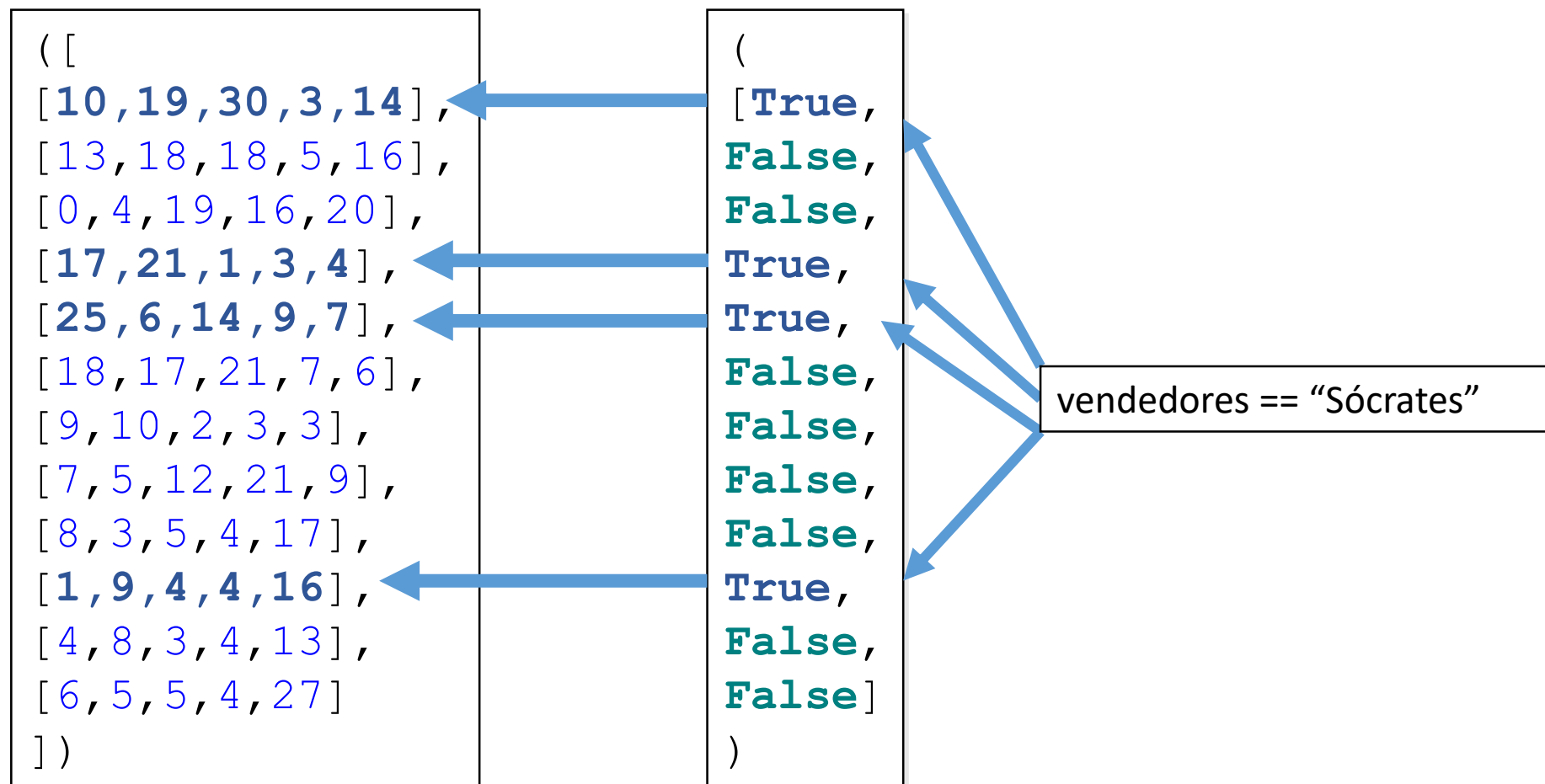
## Indexação booleana

De posse destes dois dados, como podemos usar indexação booleana para nos ajudar a selecionar as vendas de um determinado vendedor?

Resp.: Podemos criar um novo array contendo valores True e False de acordo com cada ocorrência do nome do vendedor no array que possui os nomes dos vendedores. Vamos ver como ficaria se quiséssemos somente os dados do vendedor “Sócrates”.

# NumPy

## Indexação booleana



Vamos agora implementar o exemplo.

```
import numpy as np

vendedores = np.array(["Sócrates", "Aristóteles",
                        "Platão", "Sócrates",
                        "Sócrates", "Platão",
                        "Platão", "Platão",
                        "Aristóteles", "Sócrates",
                        "Aristóteles", "Aristóteles"])

print(vendedores)
```

## Indexação booleana

```
vendas = np.array([
    [10, 19, 30, 3, 14],
    [13, 18, 18, 5, 16],
    [0, 4, 19, 16, 20],
    [17, 21, 1, 3, 4],
    [25, 6, 14, 9, 7],
    [18, 17, 21, 7, 6],
    [9, 10, 2, 3, 3],
    [7, 5, 12, 21, 9],
    [8, 3, 5, 4, 17],
    [1, 9, 4, 4, 16],
    [4, 8, 3, 4, 13],
    [6, 5, 5, 4, 27]
])

print(vendedores == "Sócrates")
print(vendas[vendedores == "Sócrates"])
```

### Resultado:

```
04_IndexacaoBooleana\codigo>ipython exemplo1.py
['Sócrates' 'Aristóteles' 'Platão' 'Sócrates' 'Sócrates' 'Platão' 'Platão'
 'Platão' 'Aristóteles' 'Sócrates' 'Aristóteles' 'Aristóteles']
[ True False False  True  True False False False  True False False]
[[10 19 30  3 14]
 [17 21  1  3  4]
 [25  6 14  9  7]
 [ 1  9  4  4 16]]
```

O array booleano deve ter o mesmo tamanho do eixo do array que ele está indexando. Podemos até misturar e fazer correspondência entre arrays booleanos e fatias, inteiros ou sequências de inteiros.

Vamos agora utilizar fatias em conjunto com o array booleano para selecionar todas as vendas do vendedor “Sócrates” que foram realizadas em dias de terça-feira.



# NumPy

## Indexação booleana

```
import numpy as np

vendedores = np.array(["Sócrates", "Aristóteles",
                       "Platão", "Sócrates",
                       "Sócrates", "Platão",
                       "Platão", "Platão",
                       "Aristóteles", "Sócrates",
                       "Aristóteles", "Aristóteles"])

print(vendedores)
```

Resultado:

```
04_IndexacaoBooleana\codigo>ipython exemplo2.py
['Sócrates' 'Aristóteles' 'Platão' 'Sócrates' 'Sócrates' 'Platão' 'Platão'
 'Platão' 'Aristóteles' 'Sócrates' 'Aristóteles' 'Aristóteles']
[[19]
 [21]
 [ 6]
 [ 9]]
```

```
vendas = np.array([
    [10, 19, 30, 3, 14],
    [13, 18, 18, 5, 16],
    [0, 4, 19, 16, 20],
    [17, 21, 1, 3, 4],
    [25, 6, 14, 9, 7],
    [18, 17, 21, 7, 6],
    [9, 10, 2, 3, 3],
    [7, 5, 12, 21, 9],
    [8, 3, 5, 4, 17],
    [1, 9, 4, 4, 16],
    [4, 8, 3, 4, 13],
    [6, 5, 5, 4, 27]
])
```

```
print(vendas[vendedores == "Sócrates", 1:2])
```

## Indexação booleana

Para retornar todos, exceto “Sócrates” podemos usar:

*`vendas[vendedores != "Sócrates"]`*

Ou negar a condição desta forma:

*`vendas[~(vendedores == "Sócrates ")]`*

O operador “~” inverte uma condição.

## Indexação booleana

Podemos também utilizar os operadores matemáticos booleanos `&` (e) e `|` (ou). As palavras reservadas ***and*** e ***or*** não funcionam com arrays booleanos.

```
validacao = (vendedores == "Sócrates") | (vendedores == "Platão")  
vendas[(validacao)]
```

Selecionar dados de um array com indexação booleana sempre criará uma cópia dos dados, mesmo que o array devolvido não seja alterado.

## Indexação booleana

No exemplo abaixo, os números 9 serão substituídos por 0 mas somente em “resultado”, não em vendas. Porque resultado é uma cópia.

```
resultado = vendas[vendedores == "Sócrates"]  
resultado[resultado == 9] = 0
```

### Outro exemplo:

*`vendas[vendedores == "Aristóteles"] = 10`*

Neste caso, todos os dados referentes a “Aristóteles” serão alterados para o valor 10.

```
['Sócrates' 'Aristóteles' 'Platão' 'Sócrates' 'Sócrates' 'Platão' 'Platão' 'Platão' 'Aristóteles' 'Sócrates' 'Aristóteles' 'Aristóteles']
```

```
Antes: [[13 18 18 5 16]
```

```
 [ 8 3 5 4 17]
```

```
 [ 4 8 3 4 13]
```

```
 [ 6 5 5 4 27]]
```

```
Depois: [[10 10 10 10 10]
```

```
 [10 10 10 10 10]
```

```
 [10 10 10 10 10]
```

```
 [10 10 10 10 10]]
```

# FIM