

pandas

Introdução às estruturas de dados do pandas

DataFrame

Estruturas de dados do pandas - DataFrame

Um DataFrame representa uma tabela de dados retangular e contém uma coleção ordenada de colunas, que podem ter valores de diferentes tipos de dados. Você pode comparar com uma planilha do Excel ou uma tabela de um banco de dados. Ele possui índice para as linhas e para as colunas. É comumente o objeto pandas mais utilizado.

pandas

Estruturas de dados do pandas - DataFrame

Existem diversas maneiras de construir um DataFrame, mas uma das mais comuns é através de um dicionário de listas de mesmo tamanho ou de arrays NumPy.

```
In [1]: import pandas as pd

In [2]: dados = {'estado': ['Minas Gerais', 'Espírito Santo', 'Rio de Janeiro', 'São Paulo'], 'populacao_2000': [17891494
...: , 3097232, 14391282, 37032403], 'populacao_2010': [19595309, 3512672, 15993583, 41252160]}

In [3]: frame = pd.DataFrame(dados)

In [4]: frame
Out[4]:
```

| | estado | populacao_2000 | populacao_2010 |
|---|----------------|----------------|----------------|
| 0 | Minas Gerais | 17891494 | 19595309 |
| 1 | Espírito Santo | 3097232 | 3512672 |
| 2 | Rio de Janeiro | 14391282 | 15993583 |
| 3 | São Paulo | 37032403 | 41252160 |

pandas

Estruturas de dados do pandas - DataFrame

Ao executar no Jupyter Notebook os objetos DataFrame serão exibidos como uma tabela HTML. Desta forma, a exibição das informações é bem melhor.

```
In [1]: import pandas as pd
```

```
In [2]: dados = {'estado': ['Minas Gerais', 'Espírito Santo', 'Rio de Janeiro', 'São Paulo'], 'populacao_2000': [17891494, 3097232, 14391282, 37032403], 'populacao_2010': [19595309, 3512672, 15993583, 41252160]}
```

```
In [3]: frame = pd.DataFrame(dados)
```

```
In [4]: frame
```

Out[4]:

| | estado | populacao_2000 | populacao_2010 |
|---|----------------|----------------|----------------|
| 0 | Minas Gerais | 17891494 | 19595309 |
| 1 | Espírito Santo | 3097232 | 3512672 |
| 2 | Rio de Janeiro | 14391282 | 15993583 |
| 3 | São Paulo | 37032403 | 41252160 |

Estruturas de dados do pandas - DataFrame

É possível modificar a ordem de exibição das colunas através do parâmetro `columns`.

```
In [5]: frame = pd.DataFrame(dados, columns=['populacao_2010', 'populacao_2000', 'estado'])
```

```
In [6]: frame
```

```
Out[6]:
```

| | populacao_2010 | populacao_2000 | estado |
|---|----------------|----------------|----------------|
| 0 | 19595309 | 17891494 | Minas Gerais |
| 1 | 3512672 | 3097232 | Espírito Santo |
| 2 | 15993583 | 14391282 | Rio de Janeiro |
| 3 | 41252160 | 37032403 | São Paulo |

Estruturas de dados do pandas - DataFrame

Se informarmos uma coluna inexistente, será exibido “valores ausentes” para esta coluna.

```
In [7]: frame = pd.DataFrame(dados, columns=['estado', 'populacao_2000', 'populacao_2010', 'populacao_2019'])
```

```
In [8]: frame
```

```
Out[8]:
```

| | estado | populacao_2000 | populacao_2010 | populacao_2019 |
|---|----------------|----------------|----------------|----------------|
| 0 | Minas Gerais | 17891494 | 19595309 | NaN |
| 1 | Espírito Santo | 3097232 | 3512672 | NaN |
| 2 | Rio de Janeiro | 14391282 | 15993583 | NaN |
| 3 | São Paulo | 37032403 | 41252160 | NaN |

Estruturas de dados do pandas - DataFrame

Podemos alterar o valor de uma coluna por atribuição. Como exemplo vamos alterar o valor da coluna `populacao_2019`.

```
In [14]: frame = pd.DataFrame(dados, columns=['estado', 'populacao_2000', 'populacao_2010', 'populacao_2019'])
```

```
In [15]: frame
```

Out[15]:

| | estado | populacao_2000 | populacao_2010 | populacao_2019 |
|---|----------------|----------------|----------------|----------------|
| 0 | Minas Gerais | 17891494 | 19595309 | NaN |
| 1 | Espírito Santo | 3097232 | 3512672 | NaN |
| 2 | Rio de Janeiro | 14391282 | 15993583 | NaN |
| 3 | São Paulo | 37032403 | 41252160 | NaN |

```
In [16]: frame['populacao_2019'] = 123456789
```

```
In [17]: frame
```

Out[17]:

| | estado | populacao_2000 | populacao_2010 | populacao_2019 |
|---|----------------|----------------|----------------|----------------|
| 0 | Minas Gerais | 17891494 | 19595309 | 123456789 |
| 1 | Espírito Santo | 3097232 | 3512672 | 123456789 |
| 2 | Rio de Janeiro | 14391282 | 15993583 | 123456789 |
| 3 | São Paulo | 37032403 | 41252160 | 123456789 |

```
In [18]: import numpy as np
```

```
In [21]: frame['populacao_2019'] = np.arange(4.)
```

```
In [22]: frame
```

Out[22]:

| | estado | populacao_2000 | populacao_2010 | populacao_2019 |
|---|----------------|----------------|----------------|----------------|
| 0 | Minas Gerais | 17891494 | 19595309 | 0.0 |
| 1 | Espírito Santo | 3097232 | 3512672 | 1.0 |
| 2 | Rio de Janeiro | 14391282 | 15993583 | 2.0 |
| 3 | São Paulo | 37032403 | 41252160 | 3.0 |

pandas

Estruturas de dados do pandas - DataFrame

Podemos obter dados de uma coluna do DataFrame usando uma notação do tipo dicionário ou por meio de atributo.

```
In [10]: frame['estado']
```

```
Out[10]: 0    Minas Gerais  
         1    Espírito Santo  
         2    Rio de Janeiro  
         3    São Paulo  
         Name: estado, dtype: object
```

```
In [11]: frame.estado
```

```
Out[11]: 0    Minas Gerais  
         1    Espírito Santo  
         2    Rio de Janeiro  
         3    São Paulo  
         Name: estado, dtype: object
```

`frame['estado']` funciona em qualquer caso, mas `frame.estado` somente funciona se o nome da coluna for um nome de variável aceita pelo Python.

Estruturas de dados do pandas - DataFrame

Podemos definir também os rótulos das linhas usando o index.

```
In [44]: frame = pd.DataFrame(dados, columns=['estado', 'populacao_2000', 'populacao_2010'],  
                             index=['um', 'dois', 'três', 'quatro'])
```

```
In [45]: frame
```

```
Out[45]:
```

| | estado | populacao_2000 | populacao_2010 |
|--------|----------------|----------------|----------------|
| um | Minas Gerais | 17891494 | 19595309 |
| dois | Espírito Santo | 3097232 | 3512672 |
| três | Rio de Janeiro | 14391282 | 15993583 |
| quatro | São Paulo | 37032403 | 41252160 |

Estruturas de dados do pandas - DataFrame

Podemos obter uma linha do DataFrame através de seu nome, utilizando o atributo especial `loc`.

```
In [4]: frame
```

```
Out[4]:
```

| | estado | populacao_2000 | populacao_2010 |
|--------|----------------|----------------|----------------|
| Um | Minas Gerais | 17891494 | 19595309 |
| Dois | Espírito Santo | 3097232 | 3512672 |
| Três | Rio de Janeiro | 14391282 | 15993583 |
| Quatro | São Paulo | 37032403 | 41252160 |

```
In [6]: frame.loc['Três']
```

```
Out[6]: estado          Rio de Janeiro
populacao_2000      14391282
populacao_2010      15993583
Name: Três, dtype: object
```

Estruturas de dados do pandas - DataFrame

Criando uma coluna nova no DataFrame utilizando uma Series. Quando atribuímos valores a uma coluna inexistente, a coluna é criada.

```
In [43]: populacao_2015 = pd.Series([21054554, 3856987, 18658754, 45145698], index=['Um', 'Dois', 'Três', 'Quatro'])
```

```
In [44]: frame['populacao_2015'] = populacao_2015
```

```
In [45]: frame
```

Out[45]:

| | estado | populacao_2000 | populacao_2010 | populacao_2015 |
|--------|----------------|----------------|----------------|----------------|
| Um | Minas Gerais | 17891494 | 19595309 | 21054554 |
| Dois | Espírito Santo | 3097232 | 3512672 | 3856987 |
| Três | Rio de Janeiro | 14391282 | 15993583 | 18658754 |
| Quatro | São Paulo | 37032403 | 41252160 | 45145698 |

Estruturas de dados do pandas - DataFrame

Podemos apagar uma coluna do DataFrame utilizando “del”.

```
In [45]: frame
```

```
Out[45]:
```

| | estado | populacao_2000 | populacao_2010 | populacao_2015 |
|--------|----------------|----------------|----------------|----------------|
| Um | Minas Gerais | 17891494 | 19595309 | 21054554 |
| Dois | Espírito Santo | 3097232 | 3512672 | 3856987 |
| Três | Rio de Janeiro | 14391282 | 15993583 | 18658754 |
| Quatro | São Paulo | 37032403 | 41252160 | 45145698 |

```
In [46]: del frame['populacao_2015']
```

```
In [47]: frame
```

```
Out[47]:
```

| | estado | populacao_2000 | populacao_2010 |
|--------|----------------|----------------|----------------|
| Um | Minas Gerais | 17891494 | 19595309 |
| Dois | Espírito Santo | 3097232 | 3512672 |
| Três | Rio de Janeiro | 14391282 | 15993583 |
| Quatro | São Paulo | 37032403 | 41252160 |

Estruturas de dados do pandas - DataFrame

Se um dicionário aninhado for passado para o DataFrame, o pandas interpretará as chaves do dicionário mais externo como as colunas e as chaves mais internas como os índices das linhas.

```
In [17]: populacao = {'Minas Gerais': {2000:17891494, 2010:19595309}, 'Espírito Santo': {2000:3097232, 2010:3512672},  
                    'Rio de Janeiro': {2000:14391282, 2010:15993583}, 'São Paulo': {2000:37032403, 2010:41252160}}
```

```
In [15]: frame_populacao = pd.DataFrame(populacao)
```

```
In [16]: frame_populacao
```

Out[16]:

| | Minas Gerais | Espírito Santo | Rio de Janeiro | São Paulo |
|------|--------------|----------------|----------------|-----------|
| 2000 | 17891494 | 3097232 | 14391282 | 37032403 |
| 2010 | 19595309 | 3512672 | 15993583 | 41252160 |

Estruturas de dados do pandas - DataFrame

Se um dicionário aninhado for passado para o DataFrame, o pandas interpretará as chaves do dicionário mais externo como as colunas e as chaves mais internas como os índices das linhas.

```
In [17]: populacao = {'Minas Gerais': {2000:17891494, 2010:19595309}, 'Espírito Santo': {2000:3097232, 2010:3512672},  
                    'Rio de Janeiro': {2000:14391282, 2010:15993583}, 'São Paulo': {2000:37032403, 2010:41252160}}
```

```
In [15]: frame_populacao = pd.DataFrame(populacao)
```

```
In [16]: frame_populacao
```

Out[16]:

| | Minas Gerais | Espírito Santo | Rio de Janeiro | São Paulo |
|------|--------------|----------------|----------------|-----------|
| 2000 | 17891494 | 3097232 | 14391282 | 37032403 |
| 2010 | 19595309 | 3512672 | 15993583 | 41252160 |

Estruturas de dados do pandas - DataFrame

Podemos também definir os atributos name para o índice e colunas.

```
In [63]: frame_populacao.index.name = 'Ano'; frame_populacao.columns.name = 'Estado'
```

```
In [64]: frame_populacao
```

```
Out[64]:
```

| Estado | Minas Gerais | Espírito Santo | Rio de Janeiro | São Paulo |
|--------|--------------|----------------|----------------|-----------|
| Ano | | | | |
| 2000 | 17891494 | 3097232 | 14391282 | 37032403 |
| 2010 | 19595309 | 3512672 | 15993583 | 41252160 |

Estruturas de dados do pandas - DataFrame

Podemos fazer a transposição do DataFrame com uma sintaxe semelhante àquela usada em um array NumPy.

```
In [67]: frame_populacao.T
```

```
Out[67]:
```

| Ano | 2000 | 2010 |
|----------------|----------|----------|
| Estado | | |
| Minas Gerais | 17891494 | 19595309 |
| Espírito Santo | 3097232 | 3512672 |
| Rio de Janeiro | 14391282 | 15993583 |
| São Paulo | 37032403 | 41252160 |

Estruturas de dados do pandas - DataFrame

Similar a Series, o atributo `values` devolve os dados contidos no DataFrame como um `ndarray` bidimensional.

```
In [31]: frame_populacao.values
```

```
Out[31]: array([[17891494,  3097232, 14391282, 37032403],  
                [19595309,  3512672, 15993583, 41252160]], dtype=int64)
```

Estruturas de dados do pandas - DataFrame

Realizando alguns cálculos simples

```
In [32]: frame
```

```
Out[32]:
```

| | estado | populacao_2000 | populacao_2010 |
|--------|----------------|----------------|----------------|
| Um | Minas Gerais | 17891494 | 19595309 |
| Dois | Espírito Santo | 3097232 | 3512672 |
| Três | Rio de Janeiro | 14391282 | 15993583 |
| Quatro | São Paulo | 37032403 | 41252160 |

```
In [33]: frame['crescimento'] = frame['populacao_2010'] - frame['populacao_2000']
```

```
In [34]: frame
```

```
Out[34]:
```

| | estado | populacao_2000 | populacao_2010 | crescimento |
|--------|----------------|----------------|----------------|-------------|
| Um | Minas Gerais | 17891494 | 19595309 | 1703815 |
| Dois | Espírito Santo | 3097232 | 3512672 | 415440 |
| Três | Rio de Janeiro | 14391282 | 15993583 | 1602301 |
| Quatro | São Paulo | 37032403 | 41252160 | 4219757 |

```
In [35]: sum(frame['populacao_2000'])
```

```
Out[35]: 72412411
```

FIM