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.

#### 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], 'população 2010': [19595309, 3512672, 15993583, 41252160]}
In [3]: frame = pd.DataFrame(dados)
In [4]: frame
Out[4]:
           estado populacao_2000 populacao_2010
    Minas Gerais
                         17891494
                                         19595309
   Espírito Santo
                         3097232
                                          3512672
   Rio de Janeiro
                        14391282
                                         15993583
       São Paulo
                         37032403
                                         41252160
```

#### 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'], 'população 2000': [17891494, 3097232, 14393
        frame = pd.DataFrame(dados)
In [4]: frame
Out[4]:
                  estado população 2000 população 2010
             Minas Gerais
                               17891494
                                              19595309
            Espírito Santo
                                               3512672
                                3097232
          2 Rio de Janeiro
                               14391282
                                              15993583
                São Paulo
                               37032403
                                              41252160
```



#### Estruturas de dados do pandas - DataFrame

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

]: fr	rame = pd.DataF	rame(dados, co	lumns=['popu
fı	rame		
	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

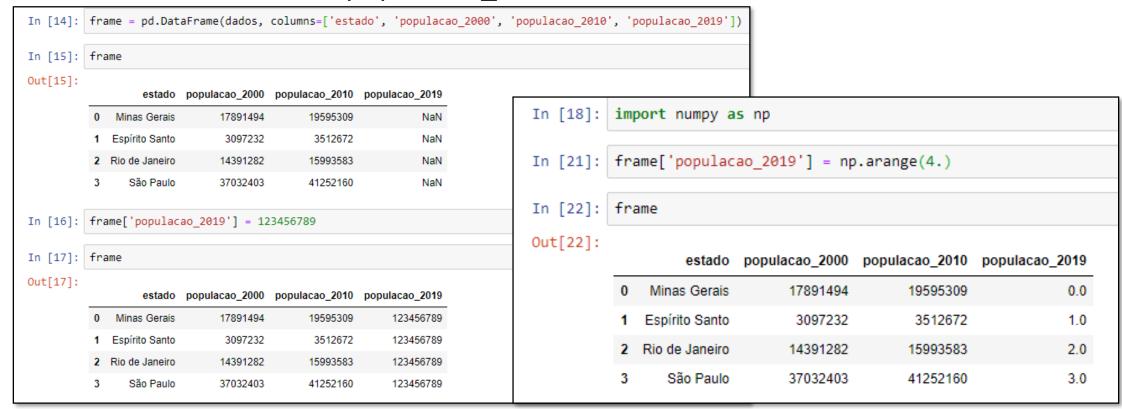
#### Estruturas de dados do pandas - DataFrame

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

In [7]:	fra	ame = pd.Dat	aFrame(dados,	columns=['esta	ado', 'populaca	o_2000', 'populacao_2010', 'populacao_20
[8]:	fra	ame				
it[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 população\_2019.





#### 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.

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 população 2000 população 2010
                   Minas Gerais
                                                     19595309
                                     17891494
              um
                   Espírito Santo
                                      3097232
                                                      3512672
             dois
             três Rio de Janeiro
                                     14391282
                                                     15993583
                     São Paulo
                                     37032403
                                                     41252160
           quatro
```



#### 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]:	populac populac	:ao_2000 :ao_2010 :rês, dtype:	Rio de Janeiro 14391282 15993583 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.

```
populacao 2015 = pd.Series([21054554, 3856987, 18658754, 45145698], index=['Um', 'Dois', 'Três', 'Quatro'])
In [43]:
          frame['populacao 2015'] = populacao 2015
In [45]:
Out[45]:
                               população 2000 população 2010 população 2015
                    Minas Gerais
                                                                    21054554
                                      17891494
                                                     19595309
                   Espírito Santo
                                       3097232
                                                      3512672
                                                                     3856987
             Três Rio de Janeiro
                                                                     18658754
                                      14391282
                                                     15993583
                      São Paulo
                                      37032403
                                                     41252160
                                                                    45145698
           Quatro
```

#### 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 fra	ame['populaca	ao_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.

```
população = {'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 população = pd.DataFrame(população)
In [16]:
         frame população
Out[16]:
                Minas Gerais Espírito Santo Rio de Janeiro São Paulo
          2000
                                3097232
                                             14391282 37032403
                   17891494
          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.

```
população = {'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 população = pd.DataFrame(população)
In [16]:
         frame população
Out[16]:
                Minas Gerais Espírito Santo Rio de Janeiro São Paulo
          2000
                                3097232
                                             14391282 37032403
                   17891494
          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	



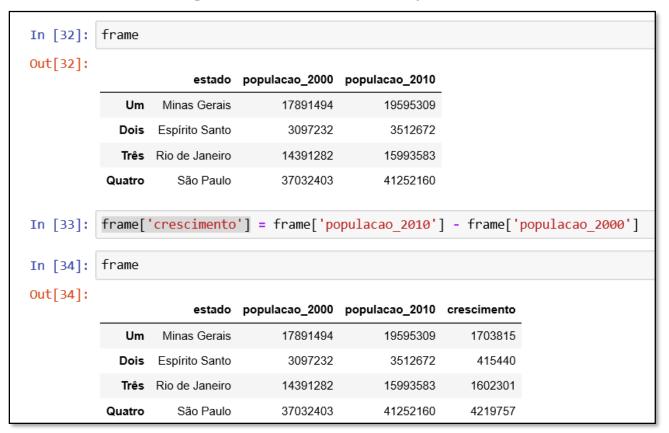


#### Estruturas de dados do pandas - DataFrame

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

#### Estruturas de dados do pandas - DataFrame

#### Realizando alguns cálculos simples



```
In [35]: sum(frame['populacao 2000'])
Out[35]: 72412411
```





# FIM

