

Pandas: Indexação, Seleção e atribuição

Professor: **Nator Junior Carvalho da Costa**

September 24, 2022

- 1 Pandas;
- 2 Series;
- 3 Atividade Prática.
- 4 DataFrame;
- 5 Atividade Prática;
- 6 LOC, ILOC, EX;
- 7 Atividade Prática.



- Pandas é uma biblioteca Python para análise de dados.

Instalação da biblioteca via PyPI:

```
1 pip install pandas
```

Iniciando com Pandas

Series



- Para utilizar a biblioteca basta realizar a importação

Instalação da biblioteca via PyPI:

```
1 # Forma tradicional de importar a biblioteca pandas
2 import pandas as pd
```

Series



Figura 1: Classe Series do Pandas.

Construindo séries a partir de um dicionário

```
1 >>> d = {'a': 1, 'b': 2, 'c': 3}
2 >>> ser = pd.Series(data=d)
3 >>> ser
4 a    1
5 b    2
6 c    3
7 dtype: int64
```

Operações com Series

Series

Construindo séries a partir de uma lista.

```
1 >>> d = [4,6,3,7]
2 >>> ser = pd.Series(data=d)
```

Operações:

- Para calcular a média, **ser.mean()**;
- Para calcular a mediana, **ser.median()**;
- Para calcular a variância, **ser.var()**;
- Para calcular a desvio padrão, **ser.std()**.

Series: Mesmo tipo de dados

Series

```
1 >>> import pandas as pd
2 >>> serie = pd.Series([42, 99, -1])
3 >>> serie
4 0    42
5 1    99
6 2    -1
7 dtype: int64
```


Series: Tipos diferentes de dados

Series

```
1 >>> serie2 = pd.Series(['hello', 2.3, True])
2 >>> serie2
3 0      hello
4 1      2.3
5 2      True
6 dtype: object
```

Acessando elementos

Series

```
1 >>> serie[0]
2 42
3 >>> serie[2]
4 -1
```

Criando índices personalizados

Series

```
1 >>> import pandas as pd
2 >>> serie = pd.Series(
3     [200, 350, 550],
4     index=['banana', 'prato feito', 'big mac']
5 )
6 >>> serie
7 banana          200
8 prato feito     350
9 big mac         550
10 dtype: int64
```

Acessando elementos

Series

```
1 >>> serie['big mac']  
2 550
```

```
1 >>> import pandas as pd
2 >>> serie = pd.Series(
3     [200, 350, 550],
4     index=[(0, 0), (0, 1), (0, 2)]
5 )
6 >>> serie
7 (0, 0)    200
8 (0, 1)    350
9 (0, 2)    550
10 dtype: int64
11 >>> serie[(0,1)]
12 350
```

Atividade 01

Series

Determine a média e variância das notas finais de 12 alunos de uma aula de Química. As notas são apresentadas na tabela abaixo e faça um programa que imprima o resultado das operações.

$$\begin{bmatrix} 8.0 & 9.1 & 7.5 & 7.4 & 7.6 & 7.8 & 8.0 & 9.3 & 7.0 & 7.0 \\ 7.2 & 7.5 & & & & & & & & \end{bmatrix} \quad (1)$$

Construindo séries a partir de um dicionário

```
1 >>> d = [8.0, 9.1, 7.5, 7.4, 7.6, 7.8, 8.0, 9.3, 7.0, 7.0, 7.2, 7.5]
2 >>> ser = pd.Series(data=d)
3 >>> print("A média é:", ser.mean())
4 >>> print("A mediana é:", ser.median())
5 >>> print("A variância é:", ser.var())
6 >>> print("A desvio padrão é:", ser.std())
```

DataFrame

The diagram illustrates the structure of a Pandas DataFrame. It features a table with two columns and five rows. The columns are labeled 'NOME' (red background) and 'IDADE' (green background). The rows are indexed from 0 to 4. A bracket on the left labeled 'Índices' points to the row indices. A bracket at the top labeled 'Colunas' points to the column headers. A bracket on the right labeled 'Valores' points to the data cells. The data is as follows:

	Colunas	
	NOME	IDADE
0	John	22
1	Maria	27
2	Joao	25
3	Paulo	35
4	Ana	40

Figura 2: Classe DataFrame do Pandas.

DataFrame

DataFrame

Criando um DataFrame

```
1 >>> import pandas as pd
2 >>> df = pd.DataFrame({
3     'calorias':[200, 350, 550],
4     'gordura (%)':[0, 6, 15]
5 })
6 >>> df
7      calorias  gordura (%)
8 0          200           0
9 1          350           6
10 2          550          15
```

DataFrame

DataFrame

Criando um DataFrame e definindo um índice.

```
1 >>> import pandas as pd
2 >>> df = pd.DataFrame({
3     'calorias':[200, 350, 550],
4     'gordura (%)':[0, 6, 15]
5 },
6   index=['banana', 'prato feito', 'big mac']
7 )
8 >>> df
9           calorias  gordura (%)
10 banana           200           0
11 prato feito       350           6
12 big mac           550          15
```

Acessando os valores de uma coluna do DataFrame.

```
1 >>> df['gordura (%)']  
2 banana          0  
3 prato feito     6  
4 big mac         15  
5 Name: gordura (%), dtype: int64  
6
```

DataFrame

DataFrame

A forma de acessar um dado de um DataFrame por meio de índices é a seguinte:

```
1 dataframe[<coluna>][<linha>]
```

Exemplo:

```
1 dados = [('Ana', 21), ('Bruno', 20), ('Carla', 22)]
2 df = pd.DataFrame(data = dados)
3 print(df)
4 #      0  1
5 #0   Ana  21
6 #1 Bruno  20
7 #2 Carla  22
8 print(df[0][0], df[0][1], df[0][2]) # Ana Bruno Carla
```

DataFrame

DataFrame

Acessando os valores de uma linha do DataFrame.

```
1 >>> df = pd.DataFrame({
2     'calorias':[200, 350, 550],
3     'gordura (%)':[0, 6, 15]
4 })
5 >>> df
6 banana          0
7 prato feito     6
8 big mac         15
9 Name: gordura (%), dtype: int64
10 >>> df[1:2]
11      calorias  gordura (%)
12 1         350          6
13
```

```
1 #podemos chamar uma linha pelo seu índice
2 df.loc[5]
3 #ou com um array de índices
4 df.loc[[0,1,2]]
5
6 # uma fatia, do quarto ao sétimo elemento
7 #(note que diferente do python puro, neste método
8 #a chave inicial e final estarão presente no resultado)
9 df.loc[4:8]
```

LOC

DataFrame

```
1 #busca simples
2 df.loc[(df['calorias']) >= 200]
3 #busca composta
4 df.loc[(df['calorias']) >= 10)&(df['calorias'] <= 350)]
5
```

```
1 #Redefinindo valores.  
2 df.loc[(df['calorias']) >= 200, 'calorias'] = 210
```



```
1 # Linhas:  
2 df.iloc[0] # Selecionado a primeira linha do dataset  
3 df.iloc[-1] # Selecionando a última linha  
4
```

```
1 # Colunas:
2 # Todos os dados da primeira coluna do dataset
3 df.iloc[:,0]
4 # Do primeiro ao quinto dado da última coluna
5 df.iloc[0:5,-1]
6
```

```
1 # Seleção de múltiplas linhas e colunas:  
2 # resgatando as primeiras três linhas do dataset  
3 df.iloc[0:3]  
4 # todos os dados da segunda e terceira coluna  
5 df.iloc[:, 1:3]  
6 # 1º, 3º e 5º elementos e 6ª a 8ª colunas  
7 df.iloc[[0,2,4], 5:8]
```

Obrigado!