

Métodos de Interpolação

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

Criando um banco qualquer

```
In [2]: data = [0.5, None, None, 0.52, 0.54, None, None, 0.59, 0.6, None, 0.7]
```

Criando uma Series com esses dados

```
In [3]: s = pd.Series(data)
s
```

```
Out[3]: 0    0.50
1     NaN
2     NaN
3    0.52
4    0.54
5     NaN
6     NaN
7    0.59
8    0.60
9     NaN
10   0.70
dtype: float64
```

Prreenchendo os valores nulos com zero

```
In [7]: # pode-se adicionar qualquer valor dentro dos parênteses, até mesmo strings
s.fillna(0)
```

```
Out[7]: 0    0.50
1    0.00
2    0.00
3    0.52
4    0.54
5    0.00
6    0.00
7    0.59
8    0.60
9    0.00
10   0.70
dtype: float64
```

Outros Métodos de Interpolação

```
In [9]: # observando a series original
s
```

```
Out[9]: 0    0.50
1     NaN
2     NaN
3    0.52
4    0.54
```

```
5      NaN
6      NaN
7      0.59
8      0.60
9      NaN
10     0.70
dtype: float64
```

Método ffill

- analisa a Series do primeiro elemento ao último
- preenche o registro nulo com o valor anteriormente válido
- método que pode ser usado com dados de uma série temporal
- quando há um valor nulo, as vezes é razoável preencher com o do ano anterior por exemplo
- isso permite que as funções sejam executadas sem problemas

```
In [10]: # esse método preenche o registro nulo com o valor anteriormente válido
s.fillna(method = 'ffill')
```

```
Out[10]: 0      0.50
1      0.50
2      0.50
3      0.52
4      0.54
5      0.54
6      0.54
7      0.59
8      0.60
9      0.60
10     0.70
dtype: float64
```

Método bfill

- funciona exatamente como o método ffill, mas ao contrário
- percorre do último ao primeiro elemento
- preenche entradas nulas com o valor válido anterior
- nesse caso o valor anterior é o valor de baixo

```
In [11]: s.fillna(method = 'bfill')
```

```
Out[11]: 0      0.50
1      0.52
2      0.52
3      0.52
4      0.54
5      0.59
6      0.59
7      0.59
8      0.60
9      0.70
10     0.70
dtype: float64
```

Preenchendo com a média de valores válidos

```
In [9]: # observando a series original
s
```

```
Out[9]: 0    0.50
        1    NaN
        2    NaN
        3    0.52
        4    0.54
        5    NaN
        6    NaN
        7    0.59
        8    0.60
        9    NaN
       10    0.70
        dtype: float64
```

```
In [12]: s.fillna(s.mean())
```

```
Out[12]: 0    0.500
        1    0.575
        2    0.575
        3    0.520
        4    0.540
        5    0.575
        6    0.575
        7    0.590
        8    0.600
        9    0.575
       10    0.700
        dtype: float64
```

Observações

- algumas vezes esses métodos de interpolação podem gerar gráficos discrepantes
- principalmente gráficos representados por curvas, podem gerar retas, onde há a interpolação
- por isso pode-se limitar o nº de preenchimentos com o parâmetro 'limit'
- também é possível mesclar os métodos para suavizar esse impacto

Limitando a interpolação

```
In [9]: # observando a series original
        s
```

```
Out[9]: 0    0.50
        1    NaN
        2    NaN
        3    0.52
        4    0.54
        5    NaN
        6    NaN
        7    0.59
        8    0.60
        9    NaN
       10    0.70
        dtype: float64
```

```
In [14]: # limitando a um preenchimento
        s.fillna(method = 'ffill', limit = 1)
```

```
Out[14]: 0    0.50
        1    0.50
```

```
2      NaN
3      0.52
4      0.54
5      0.54
6      NaN
7      0.59
8      0.60
9      0.60
10     0.70
dtype: float64
```

Mesclando os Métodos

- mesclando ffill e bfill
- dessa forma suavizamos os dados, preenchendo 1º com os dados acima e depois com os dados válidos de baixo

```
In [15]: s1 = s.fillna(method = 'ffill', limit = 1)
s1
```

```
Out[15]: 0      0.50
1      0.50
2      NaN
3      0.52
4      0.54
5      0.54
6      NaN
7      0.59
8      0.60
9      0.60
10     0.70
dtype: float64
```

```
In [16]: s1.fillna(method = 'bfill', limit = 1)
```

```
Out[16]: 0      0.50
1      0.50
2      0.52
3      0.52
4      0.54
5      0.54
6      0.59
7      0.59
8      0.60
9      0.60
10     0.70
dtype: float64
```