Além das funções [highlight\_max](https://pandas.pydata.org/docs/reference/api/pandas.io.formats.style.Styler.highlight_max.html), [highlight\_min](https://pandas.pydata.org/docs/reference/api/pandas.io.formats.style.Styler.highlight_min.html), [background\_gradient](https://pandas.pydata.org/docs/reference/api/pandas.io.formats.style.Styler.background_gradient.html) e [bar](https://pandas.pydata.org/docs/reference/api/pandas.io.formats.style.Styler.bar.html) que conhecemos durante as aulas, existem outros estilos integrados ao Styler que podemos aplicar na tabela.

As funções built-in permitem aplicar estilizações comuns na nossa visualização, sem a necessidade de se construir um código específico para aplicá-las na tabela. Vamos conhecer essas funções abaixo.

* [highlight\_null](https://pandas.pydata.org/docs/reference/api/pandas.io.formats.style.Styler.highlight_null.html): identifica e destaca elementos faltantes (valores nulos).
* [highlight\_between](https://pandas.pydata.org/docs/reference/api/pandas.io.formats.style.Styler.highlight_between.html): identifica e destaca elementos que estão dentro de uma faixa específica de valores.
* [highlight\_quantile](https://pandas.pydata.org/docs/reference/api/pandas.io.formats.style.Styler.highlight_quantile.html): identifica e destaca elementos de acordo com a faixa de quantil especificada.
* [text\_gradient](https://pandas.pydata.org/docs/reference/api/pandas.io.formats.style.Styler.text_gradient.html): funciona de forma similar ao background\_gradient, criando um gradiente no texto baseado na escala numérica dos valores presentes na coluna a ser transformada ou em outra.

Para exemplificar, vamos construir um código com highlight\_null e highlight\_between. Iremos utilizar uma tabela com dados salariais de 5 funcionários de uma empresa:

|  | **ID** | **Idade** | **Salário** |
| --- | --- | --- | --- |
| 0 | 1204 | nan | 3500 |
| 1 | 3018 | 32 | 4200.5 |
| 2 | 9963 | 19 | 2800 |
| 3 | 1005 | nan | 5700.25 |
| 4 | 5627 | 38 | 4900.75 |

O código do DataFrame utilizado está dado abaixo para que você consiga aplicar o exemplo:

import pandas as pd

import numpy as np

dados = pd.DataFrame({'ID': [1204, 3018, 9963, 1005, 5627],

'Idade': [np.nan, 32, 19, np.nan, 38],

'Salário': [3500.00, 4200.50, 2800.00, 5700.25, 4900.75]})

**Copiar código**

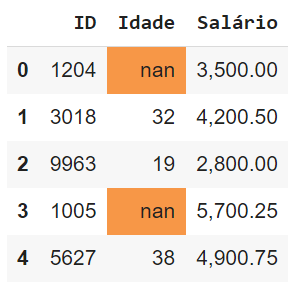
O highlight\_null será aplicado no objeto Styler, e a cor de fundo dos elemento que tem dados nulos é alterada para laranja (#F79747).

s\_nan = dados.style.format({'Idade':'{:.0f}','Salário':'{:,.2f}'})\

.highlight\_null(color='#F79747')

s\_nan

**Copiar código**

****

Para o highlight\_between, podemos definir um limite superior e inferior para destacar um valor que esteja entre esses limites. No exemplo, o objetivo é destacar os valores na coluna Salário que estão entre 4 e 5 mil. O parâmetro left representa o limite inferior e o right o limite superior dos valores.

## Definir os limites superior e inferior

limite\_superior = 5000.0

# O limite inferior é especificado através de uma Series que informa qual a coluna que terá a aplicação da **fun**ção

limite\_inferior = pd.Series([4000.0], index=['Salário'])

## Aplicar a estilização com highlight\_between

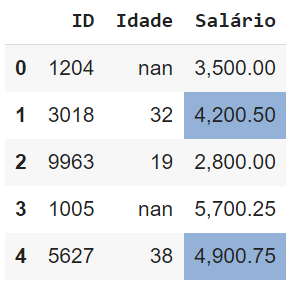
s\_bet = dados.style.format({'Idade':'{:.0f}','Salário':'{:,.2f}'})\

.highlight\_between(left=limite\_inferior, right=limite\_superior,

axis=1, color='#94B2D7')

s\_bet

**Copiar código**

****

Caso você queira utilizar os outros métodos, você pode consultar a documentação para entender como aplicá-los.