Existem diversas maneiras de tratar dados nulos com a biblioteca Pandas utilizando Python. Existe uma diversidade de funções, métodos e cálculos que conseguimos realizar com os dados faltantes através do Pandas.

No artigo [*Working with missing data* (em português, Trabalhando com dados faltantes)](https://pandas.pydata.org/docs/user_guide/missing_data.html), da biblioteca Pandas, você pode aprender tudo o que o Pandas consegue fazer com esse tipo de dado. Nessa atividade, vamos focar em algumas funções que a biblioteca disponibiliza para tratar dados faltantes.

#### **dropna()**

Este método **remove** linhas ou colunas que contenham valores nulos do DataFrame. É interessante utilizá-lo quando os dados nulos representam apenas uma pequena porcentagem do conjunto de dados ou quando esses dados não são essenciais para a análise.

#### **fillna()**

Este método permite **preencher** os valores nulos com um valor específico fornecido pelo usuário, como uma constante, a média, a mediana ou a moda dos dados existentes. Em uma situação que é necessário manter todas as linhas e colunas do DataFrame, ou quando é razoável substituir os valores nulos por um valor comum, podemos utilizar esse método tranquilamente

#### **interpolate()**

Este método preenche valores ausentes em um DataFrame usando métodos de interpolação. Interpolação é um método utilizado para estimar valores dentro de um intervalo conhecido de dados observados, assim conseguimos estimar valores ausentes com base nos dados existentes. Pode ser útil utilizá-lo quando temos dados que seguem padrões lineares ou suaves, pois a interpolação pode manter a continuidade e a coerência dos valores.

#### **rolling()**

Esse método é utilizado para calcular estatísticas resumidas ao longo de uma janela de tempo ou de uma janela específica de observações. Ele é comumente usado em séries temporais para calcular médias móveis, desvios padrão móveis, entre outras estatísticas.

É interessante utilizar esse método quando os dados estão organizados em uma sequência (pode ser temporal) e desejamos preencher valores faltantes com base em uma média móvel ou em outras estatísticas calculadas ao longo de uma janela de tempo. Por exemplo, podemos usar uma média móvel para preencher lacunas em séries temporais onde a tendência é relativamente constante ao longo do tempo.

#### **isnull()**

Este método retorna uma máscara booleana indicando os valores nulos no DataFrame. Melhor contexto: Quando você precisa identificar onde estão os valores nulos no DataFrame, pode ser usado em conjunto com outras operações.

#### **ffill()**

Esse método já conhecemos em aula; ele preenche os valores ausentes com o último valor não nulo conhecido, propagando esse valor para frente até encontrar um novo valor não nulo. Em um caso em que os dados têm uma tendência estável e a última observação válida pode ser considerada uma boa estimativa para os valores faltantes no futuro, podemos usar esse método com tranquilidade. Por exemplo, em um conjunto de dados onde os valores mudam lentamente ao longo do tempo e podemos razoavelmente supor que o próximo valor será semelhante ao último observado.

#### **bfill()**

Ao contrário do ffill(), o método bfill() preenche os valores ausentes com o próximo valor não nulo conhecido, propagando esse valor para trás até encontrar um novo valor não nulo. Por isso, ele acaba sendo útil quando os dados têm uma tendência constante e as observações seguintes podem ser uma boa estimativa para os valores faltantes no passado.