**Biblioteca Pandas Python**

**MÉTODOS DE ENTRADA**

**.read\_sql\_query**

**.read\_csv**

**.read\_fwf**

**.read\_json**

**.read\_html**

**.read\_xml**

**.read\_clipboard**

**.read\_excel**

**.read\_excel**

**.read\_hdf**

**.read\_feather**

**.read\_parquet**

**.read\_orc**

**.read\_stata**

**.read\_sas**

**.read\_spss**

**.read\_pickle**

**.read\_sql**

**.read\_gbq**

**MÉTODOS DE SAÍDA**

**.to\_csv**

**.to\_json**

**.to\_html**

**.Styler.to\_latex**

**.to\_xml**

**.to\_clipboard**

**.to\_excel**

**.to\_hdf**

**.to\_feather**

**.to\_parquet**

**.to\_stata**

**.to\_pickle**

**.to\_sql**

**.to\_gbq**

**Métodos de conferência**

**.attrs**

Dicionário de atributos globais deste conjunto de dados.

**.index**

Retorna o índice em linhas do dataframe.

**.columns**

Retorna o nome de todas as colunas do dataframe

**.dtypes**

Retorna o tipo de dados do Dataframe.

**.info(*verbose=None*, *buf=None*, *max\_cols=None*, *memory\_usage=None*, *show\_counts=None*, *null\_counts=None*)**

Retorna um sumário do dataframe

**.shape**

Retorna uma tupla com o número de linhas e colunas do df.

**.empty**

Indica se há colunas ou linhas vazias

**.head(*n=5*)**

Retorna as primeiras linhas do dataframe.

**.tail(*n=5*)**

Retorna as últimas linhas do df.

**.tail(*n=5*)**

Retorna as últimas n linhas.

**.isin(*values*)**

Retorna se cada elemento no dataframe esta contido nos valores.

**.query(*expr*, *inplace=False*, *\*\*kwargs*)**

Consulta as colunas com uma expressão booleana.

**.abs()**

Retorna um Series/DataFrame com valor numérico absoluto de cada elemento. Esta função só se aplica a elementos que são todos numéricos.

**.all(*axis=0*, *bool\_only=None*, *skipna=True*, *level=None*, *\*\*kwargs*)**

Retorna se todos os elementos são True, potencialmente sobre um eixo. Retorna True a menos que haja pelo menos um elemento dentro de uma série ou ao longo de um eixo de Dataframe que seja False ou equivalente (por exemplo, zero ou vazio).

**.any(*axis=0*, *bool\_only=None*, *skipna=True*, *level=None*, *\*\*kwargs*)**

Retorna se algum elemento é True, potencialmente sobre um eixo. Retorna False, a menos que haja pelo menos um elemento em uma série ou ao longo de um eixo do Dataframe que seja True ou equivalente (por exemplo, diferente de zero ou não vazio).

**.count(*axis=0*, *level=None*, *numeric\_only=False*)**

Conte as células não-NA para cada coluna ou linha. Os valores None, NaN, NaT e opcionalmente numpy.inf (dependendo de pandas.options.mode.use\_inf\_as\_na) são considerados NA.

**.describe(*percentiles=None*, *include=None*, *exclude=None*, *datetime\_is\_numeric=False*)**

Gera estatísticas descritivas. Estatísticas descritivas incluem aquelas que resumem a tendência central, dispersão e forma da distribuição de um conjunto de dados, excluindo os valores de NaN. Analisa séries numéricas e de objetos, bem como conjuntos de colunas DataFrame de tipos de dados mistos. A saída irá variar dependendo do que é fornecido. Consulte as notas abaixo para obter mais detalhes.

**.idxmax(*axis=0*, *skipna=True*)**

Índice de retorno da primeira ocorrência de máximo sobre o eixo solicitado. Valores NA/nulos são excluídos.

**.idxmin(*axis=0*, *skipna=True*)**

Índice de retorno da primeira ocorrência do mínimo sobre o eixo solicitado.

**.compare(*other*, *align\_axis=1*, *keep\_shape=False*, *keep\_equal=False*)**

Compare com outro DataFrame e mostre as diferenças.

**.asof(*where*, *subset=None*)**

Retorna a(s) última(s) linha(s) sem nenhum NaN antes de where.

**.infer\_objects()**

Infere melhores tipos para colunas de objetos.

**.value\_counts(*subset=None*, *normalize=False*, *sort=True*, *ascending=False*, *dropna=True*)**

Retorna uma série contendo contagens de linhas exclusivas no DataFrame.

**.nunique(*axis=0*, *dropna=True*)**

Conta o número de elementos distintos no eixo especificado. Série de retorno com número de elementos distintos. Pode ignorar valores NaN.

**.sample(*n=None*, *frac=None*, *replace=False*, *weights=None*, *random\_state=None*, *axis=None*, *ignore\_index=False*)**

Retorna uma amostra aleatória de itens de um eixo de objeto.

**MÉTODOS DE SELEÇÃO e MANIPULAÇÃO**

**.loc**

Acessa um grupo de linhas e colunas pelo rotulo ou uma array booleana.

**.iloc**

Localiza a posição pelo número inteiro passado.

**.get(*key*, *default=None*)**

Pega um item de um objeto para dada chave.

**.filter(*items=None*, *like=None*, *regex=None*, *axis=None*)**

Subconjunto as linhas ou colunas do dataframe de acordo com os rótulos de índice especificados. Observe que essa rotina não filtra um dataframe em seu conteúdo. O filtro é aplicado aos rótulos do índice.

**.insert(*loc*, *column*, *value*, *allow\_duplicates=False*)**

Insere uma coluna em uma posição específica.

**.copy(*deep=True*)**

Faz um cópia dos índices do objeto e dados.

**.iteritems()**

Itera sobre pares de colunas series (usado com for).

**.iterrows()**

Itera sobre linhas do dataframe como pares.

**.itertuples(*index=True*, *name='Pandas'*)**

Itera sobre linhas do dataframe como tuplas nomeadas.

**.pop(*item*)**

Deleta uma coluna.

**.first(*offset*)**

Seleciona períodos iniciais de dados de séries temporais com base em um deslocamento de data. Ao ter um DataFrame com datas como índice, esta função pode selecionar as primeiras linhas com base em um deslocamento de data.

**.last(*offset*)**

Seleciona períodos finais de dados de séries temporais com base em um deslocamento de data.

**.apply(*func*, *axis=0*, *raw=False*, *result\_type=None*, *args=()*, *\*\*kwargs*)**

Aplica uma função ao longo de um eixo do DataFrame.

**.applymap(*func*, *na\_action=None*, *\*\*kwargs*)**

Aplica uma função a um Dataframe elementwise. Este método aplica uma função que aceita e retorna um escalar para cada elemento de um DataFrame.

**.pipe(*func*, *\*args*, *\*\*kwargs*)**

Aplica funções encadeadas que esperam Series ou DataFrames.

**.append(*other*, *ignore\_index=False*, *verify\_integrity=False*, *sort=False*)**

Anexe linhas de outro ao final do chamador, retornando um novo objeto.

**.assign(*\*\*kwargs*)**

Atribua novas colunas a um DataFrame.

**.join(*other*, *on=None*, *how='left'*, *lsuffix=''*, *rsuffix=''*, *sort=False*)**

Junte colunas de outro DataFrame.

**.merge(*right*, *how='inner'*, *on=None*, *left\_on=None*, *right\_on=None*, *left\_index=False*, *right\_index=False*, *sort=False*, *suffixes=('\_x', '\_y')*, *copy=True*, *indicator=False*, *validate=None*)**

Mescle objetos DataFrame ou Series nomeados com uma junção de estilo de banco de dados.

**.add\_prefix(*prefix*)**

Etiquetas de prefixo com prefixo de string. Para Série, os rótulos de linha são prefixados. Para DataFrame, os rótulos de coluna são prefixados.

**.add\_suffix(*suffix*)**

Rótulos de sufixo com sufixo de string. Para Série, os rótulos de linha são sufixados. Para DataFrame, os rótulos de coluna são sufixados.

**MÉTODOS DE INDEXAÇÃO e ORDENAÇÃO**

**.reindex(*labels=None*, *index=None*, *columns=None*, *axis=None*, *method=None*, *copy=True*, *level=None*, *fill\_value=nan*, *limit=None*, *tolerance=None*)**

Conforma as Series/DataFrame ao novo índice com lógica de preenchimento opcional.

**.reindex\_like(*other*, *method=None*, *copy=True*, *limit=None*, *tolerance=None*)**

Retorna um objeto com índices correspondentes como outro objeto.

**.set\_axis(*labels*, *axis=0*, *inplace=False*)**

Atribui o índice desejado a um determinado eixo.

**.set\_index(*keys*, *drop=True*, *append=False*, *inplace=False*, *verify\_integrity=False*)**

Define o índice DataFrame usando colunas existentes.

**.take(*indices*, *axis=0*, *is\_copy=None*, *\*\*kwargs*)**

Retorna os elementos nos índices posicionais fornecidos ao longo de um eixo. Isso significa que não estamos indexando de acordo com os valores reais no atributo index do objeto. Estamos indexando de acordo com a posição real do elemento no objeto.

**.droplevel(*level*, *axis=0*)**

Retornar Series/DataFrame com índice/níveis de coluna solicitados removidos.

**.reorder\_levels(*order*, *axis=0*)**

Reorganize os níveis de índice usando a ordem de entrada. Não pode cair ou duplicar níveis.

**.sort\_values(*by*, *axis=0*, *ascending=True*, *inplace=False*, *kind='quicksort'*, *na\_position='last'*, *ignore\_index=False*, *key=None*)**

Classifique pelos valores ao longo de qualquer eixo.

**.sort\_index(*axis=0*, *level=None*, *ascending=True*, *inplace=False*, *kind='quicksort'*, *na\_position='last'*, *sort\_remaining=True*, *ignore\_index=False*, *key=None*)**

Classifique o objeto por rótulos (ao longo de um eixo). Retorna um novo DataFrame classificado por rótulo se o argumento inplace for False, caso contrário atualiza o DataFrame original e retorna None.

**.rank(*axis=0*, *method='average'*, *numeric\_only=NoDefault.no\_default*, *na\_option='keep'*, *ascending=True*, *pct=False*)**

Calcula as classificações de dados numéricos (1 a n) ao longo do eixo. Por padrão, valores iguais recebem uma classificação que é a média das classificações desses valores.

**.align(*other*, *join='outer'*, *axis=None*, *level=None*, *copy=True*, *fill\_value=None*, *method=None*, *limit=None*, *fill\_axis=0*, *broadcast\_axis=None*)**

Alinha dois objetos em seus eixos com o método de junção especificado. O método de junção é especificado para cada índice de eixo.

**.pivot(*index=None*, *columns=None*, *values=None*)**

Retornar DataFrame remodelado organizado por valores de índice/coluna fornecidos. Reformule os dados (produza uma tabela “dinâmica”) com base nos valores da coluna. Usa valores exclusivos de índice/colunas especificados para formar eixos do DataFrame resultante. Esta função não suporta agregação de dados, vários valores resultarão em um MultiIndex nas colunas. Consulte o Guia do usuário para obter mais informações sobre remodelação.

**.pivot\_table(*values=None*, *index=None*, *columns=None*, *aggfunc='mean'*, *fill\_value=None*, *margins=False*, *dropna=True*, *margins\_name='All'*, *observed=False*, *sort=True*)**

Crie uma tabela dinâmica no estilo de planilha como um DataFrame.

**MÉTODOS DE CÁLCULO e ESTATÍSTICA**

**.add(*other*, *axis='columns'*, *level=None*, *fill\_value=None*)**

Obtem adição de dataframe e outros elementos (adicionar operador binário).

**.sub(*other*, *axis='columns'*, *level=None*, *fill\_value=None*)**

Obtem subtração de dataframe e outros elementos (sub operador binário).

**.mul(*other*, *axis='columns'*, *level=None*, *fill\_value=None*)**

Obter Multiplicação de dataframe e outros elementos (operador binário mul).

**.div(*other*, *axis='columns'*, *level=None*, *fill\_value=None*)**

Obter divisão flutuante de dataframe e outros elementos (operador binário truediv).

**.truediv(*other*, *axis='columns'*, *level=None*, *fill\_value=None*)**

Obter divisão flutuante de dataframe e outros elementos (operador binário truediv).

**.floordiv(*other*, *axis='columns'*, *level=None*, *fill\_value=None*)**

Obtem a divisão Integer do dataframe e outros elementos (operador binário floordiv).

**.pow(*other*, *axis='columns'*, *level=None*, *fill\_value=None*)**

Obtem a potência exponencial do dataframe e outros elementos (operador binário pow).

**.dot(*other*)**

Calcula a multiplicação de matrizes entre o DataFrame e outro.

**.radd(*other*, *axis='columns'*, *level=None*, *fill\_value=None*)**

Obtem adição de dataframe e outros elementos (operador binário radd).

**.rsub(*other*, *axis='columns'*, *level=None*, *fill\_value=None*)**

Obtem subtração de dataframe e outros elementos (operador binário rsub).

**.rmul(*other*, *axis='columns'*, *level=None*, *fill\_value=None*)**

Obtem Multiplicação de dataframe e outros elementos (operador binário rmul).

**.rdiv(*other*, *axis='columns'*, *level=None*, *fill\_value=None*)**

Obtem divisão flutuante de dataframe e outros elementos (operador binário rtruediv).

**.rtruediv(*other*, *axis='columns'*, *level=None*, *fill\_value=None*)**

Obter divisão flutuante de dataframe e outros elementos (operador binário rtruediv).

**.rfloordiv(*other*, *axis='columns'*, *level=None*, *fill\_value=None*)**

Obtem a divisão Integer do dataframe e outros elementos (operador binário rfloordiv).

**.rolling(*window*, *min\_periods=None*, *center=False*, *win\_type=None*, *on=None*, *axis=0*, *closed=None*, *method='single'*)**

Fornece cálculos de janela rolante.

**.expanding(*min\_periods=1*, *center=None*, *axis=0*, *method='single'*)**

Fornece cálculos de janela de expansão.

**.ewm(*com=None*, *span=None*, *halflife=None*, *alpha=None*, *min\_periods=0*, *adjust=True*, *ignore\_na=False*, *axis=0*, *times=None*, *method='single'*)**

Fornece cálculos exponencialmente ponderados (EW).

**.corr(*method='pearson'*, *min\_periods=1*)**

Compute pairwise correlation of columns, excluding NA/null values.

**.corrwith(*other*, *axis=0*, *drop=False*, *method='pearson'*)**

Calcular correlação de pares. A correlação de pares é calculada entre linhas ou colunas de DataFrame com linhas ou colunas de Series ou DataFrame. Os DataFrames são primeiro alinhados ao longo de ambos os eixos antes de calcular as correlações.

**.cov(*min\_periods=None*, *ddof=1*)**

Calcula a covariância par a par de colunas, excluindo valores NA/nulos. Calcule a covariância de pares entre as séries de um DataFrame. O quadro de dados retornado é a matriz de covariância das colunas do DataFrame. Os valores NA e nulos são automaticamente excluídos do cálculo. (Consulte a nota abaixo sobre viés de valores ausentes.) Um limite pode ser definido para o número mínimo de observações para cada valor criado. As comparações com observações abaixo desse limite serão retornadas como NaN. Esse método geralmente é usado para a análise de dados de séries temporais para entender a

relação entre diferentes medidas ao longo do tempo.

**.cummax(*axis=None*, *skipna=True*, *\*args*, *\*\*kwargs*)**

Retorna o máximo cumulativo em um eixo DataFrame ou Series. Retorna um DataFrame ou Series do mesmo tamanho contendo o máximo cumulativo.

**.cummin(*axis=None*, *skipna=True*, *\*args*, *\*\*kwargs*)**

Retorna o mínimo cumulativo em um eixo DataFrame ou Series. Retorna um DataFrame ou Series do mesmo tamanho contendo o mínimo cumulativo.

**.cumprod(*axis=None*, *skipna=True*, *\*args*, *\*\*kwargs*)**

Devolve o produto cumulativo em um eixo DataFrame ou Series. Retorna um DataFrame ou Series do mesmo tamanho que contém o produto cumulativo.

**.cumsum(*axis=None*, *skipna=True*, *\*args*, *\*\*kwargs*)**

Retorna a soma cumulativa em um eixo DataFrame ou Series. Retorna um DataFrame ou Series do mesmo tamanho contendo a soma cumulativa.

**.kurt(*axis=NoDefault.no\_default*, *skipna=True*, *level=None*, *numeric\_only=None*, *\*\*kwargs*)**

Retorna a curtose imparcial sobre o eixo solicitado. Curtose obtida usando a definição de curtose de Fisher (curtose de normal == 0,0). Normalizado por N-1.

**.kurtosis(*axis=NoDefault.no\_default*, *skipna=True*, *level=None*, *numeric\_only=None*, *\*\*kwargs*)**

Retorna a curtose imparcial sobre o eixo solicitado. Curtose obtida usando a definição de curtose de Fisher (curtose de normal == 0,0). Normalizado por N-1.

**.mad(*axis=None*, *skipna=True*, *level=None*)**

Retorna o desvio absoluto médio dos valores sobre o eixo solicitado.

**.max(*axis=NoDefault.no\_default*, *skipna=True*, *level=None*, *numeric\_only=None*, *\*\*kwargs*)**

Retorna o máximo dos valores sobre o eixo solicitado. Se você quiser o índice do máximo, use idxmax. Este é o equivalente do método numpy.ndarray argmax.

**.mean(*axis=NoDefault.no\_default*, *skipna=True*, *level=None*, *numeric\_only=None*, *\*\*kwargs*)**

Retorna a média dos valores sobre o eixo solicitado.

**.median(*axis=NoDefault.no\_default*, *skipna=True*, *level=None*, *numeric\_only=None*, *\*\*kwargs*)**

Retorna a mediana dos valores sobre o eixo solicitado.

**.min(*axis=NoDefault.no\_default*, *skipna=True*, *level=None*, *numeric\_only=None*, *\*\*kwargs*)**

Retorna o mínimo dos valores sobre o eixo solicitado. Se você quiser o índice do mínimo, use idxmin. Este é o equivalente do método numpy.ndarray argmin.

**.mode(*axis=0*, *numeric\_only=False*, *dropna=True*)**

Obtem o(s) modo(s) de cada elemento ao longo do eixo selecionado. A moda de um conjunto de valores é o valor que aparece com mais frequência. Pode ser vários valores.

**.diff(*periods=1*, *axis=0*)**

Primeira diferença discreta do elemento. Calcula a diferença de um elemento Dataframe comparado com outro elemento no Dataframe (o padrão é elemento na linha anterior).

**.pct\_change(*periods=1*, *fill\_method='pad'*, *limit=None*, *freq=None*, *\*\*kwargs*)**

Altera o percentual entre o elemento atual e o anterior. Calcula a alteração percentual da linha imediatamente anterior por padrão. Isso é útil para comparar a porcentagem de mudança em uma série temporal de elementos.

**.prod(*axis=None*, *skipna=True*, *level=None*, *numeric\_only=None*, *min\_count=0*, *\*\*kwargs*)**

Retorna o produto dos valores sobre o eixo solicitado.

**.product(*axis=None*, *skipna=True*, *level=None*, *numeric\_only=None*, *min\_count=0*, *\*\*kwargs*)**

Retorna o produto dos valores sobre o eixo solicitado

**.quantile(*q=0.5*, *axis=0*, *numeric\_only=True*, *interpolation='linear'*)**

Valores de retorno no quantil fornecido sobre o eixo solicitado.

**.round(*decimals=0*, *\*args*, *\*\*kwargs*)**

Arredonda um DataFrame para um número variável de casas decimais.

**.sem(*axis=None*, *skipna=True*, *level=None*, *ddof=1*, *numeric\_only=None*, *\*\*kwargs*)**

Retorna o erro padrão imparcial da média sobre o eixo solicitado. Normalizado por N-1 por padrão. Isso pode ser alterado usando o argumento ddof

**.skew(*axis=NoDefault.no\_default*, *skipna=True*, *level=None*, *numeric\_only=None*, *\*\*kwargs*)**

Retorna a inclinação imparcial sobre o eixo solicitado. Normalizado por N-1.

**.sum(*axis=None*, *skipna=True*, *level=None*, *numeric\_only=None*, *min\_count=0*, *\*\*kwargs*)**

Retorna a soma dos valores sobre o eixo solicitado.

**.std(*axis=None*, *skipna=True*, *level=None*, *ddof=1*, *numeric\_only=None*, *\*\*kwargs*)**

Retorna o desvio padrão da amostra sobre o eixo solicitado. Normalizado por N-1 por padrão. Isso pode ser alterado usando o argumento ddof.

**.var(*axis=None*, *skipna=True*, *level=None*, *ddof=1*, *numeric\_only=None*, *\*\*kwargs*)**

Retorna a variação imparcial sobre o eixo solicitado. Normalizado por N-1 por padrão. Isso pode ser alterado usando o argumento ddof.

**MÉTODOS DE CONVERSÃO e SUBSTITUIÇÃO**

**.astype(*dtype*, *copy=True*, *errors='raise'*)**

Converte um objeto em um tipo específico.

**.convert\_dtypes(*infer\_objects=True*, *convert\_string=True*, *convert\_integer=True*, *convert\_boolean=True*, *convert\_floating=True*)**

Converte as colunas para o melhor tipo possível usando dtypes que suportam pd.na

**.where(*cond*, *other=NoDefault.no\_default*, *inplace=False*, *axis=None*, *level=None*, *errors='raise'*, *try\_cast=NoDefault.no\_default*)**

Substitui valores se a condição é falsa.

**.mask(*cond*, *other=nan*, *inplace=False*, *axis=None*, *level=None*, *errors='raise'*, *try\_cast=NoDefault.no\_default*)**

Substitui valores se a condição é verdadeira.

**.rename(*mapper=None*, *\**, *index=None*, *columns=None*, *axis=None*, *copy=True*, *inplace=False*, *level=None*, *errors='ignore'*)**

Altera os rótulos dos eixos. Os valores de função / dict devem ser exclusivos (1 para 1). Rótulos não contidos em um dict/Série serão deixados como estão. Os rótulos extras listados não geram um erro.

**.rename\_axis(*mapper=None*, *index=None*, *columns=None*, *axis=None*, *copy=True*, *inplace=False*)**

Define o nome do eixo para o índice ou colunas.

**.reset\_index(*level=None*, *drop=False*, *inplace=False*, *col\_level=0*, *col\_fill=''*)**

Redefine o índice ou um nível dele. Redefina o índice do DataFrame e use o padrão. Se o DataFrame tiver um MultiIndex, esse método poderá remover um ou mais níveis.

**.replace(*to\_replace=None*, *value=NoDefault.no\_default*, *inplace=False*, *limit=None*, *regex=False*, *method=NoDefault.no\_default*)**

Substitua os valores fornecidos em to\_replace por value.

**.transpose( )**

Troca colunas por linhas e vice e versa

**.to\_period(*freq=None*, *axis=0*, *copy=True*)**

Converta DataFrame de DatetimeIndex para PeriodIndex.

**.tz\_convert(*tz*, *axis=0*, *level=None*, *copy=True*)**

Converta o eixo tz-aware para o fuso horário de destino.

**.explode(*column*, *ignore\_index=False*)**

Transforme cada elemento de uma lista em uma linha, replicando valores de índice.

**.combine(*other*, *func*, *fill\_value=None*, *overwrite=True*)**

Combina colunas com outro DataFrame.

**.combine\_first(*other*)**

Atualiza elementos nulos com valor no mesmo local em outro.

**.swapaxes(*axis1*, *axis2*, *copy=True*)**

Troque os eixos e troque os eixos de valores apropriadamente.

**MÉTODOS LÓGICOS**

**.mod(*other*, *axis='columns'*, *level=None*, *fill\_value=None*)**

Obtem o módulo de dataframe e outros elementos (mod de operador binário).

**.rmod(*other*, *axis='columns'*, *level=None*, *fill\_value=None*)**

Obtem o módulo de dataframe e outros elementos (operador binário rmod).

**.rpow(*other*, *axis='columns'*, *level=None*, *fill\_value=None*)**

Obtem o poder exponencial do dataframe e outros elementos (operador binário rpow).

**.lt(*other*, *axis='columns'*, *level=None*)**

Obtem o ‘menor do que’ do dataframe e outros elementos (operador binário lt).

**.gt(*other*, *axis='columns'*, *level=None*)**

Obtem o ‘maior do que’ do dataframe e outros elementos (operador binário gt).

**.le(*other*, *axis='columns'*, *level=None*)**

Obtem menor ou igual a do dataframe e outros, elemento a elemento (operador binário le).

**.ge(*other*, *axis='columns'*, *level=None*)**

Obtem Maior ou igual a do dataframe e outros, elemento a elemento (operador binário ge).

**.ne(*other*, *axis='columns'*, *level=None*)**

Obtem o ‘não igual a’ do dataframe e outros, elemento a elemento (operador binário ne).

**.eq(*other*, *axis='columns'*, *level=None*)**

Obtem o ‘Igual a’ do dataframe e outros, elemento a elemento (operador binário eq).

**MÉTODOS DE AGREGAÇÃO**

**.agg(*func=None*, *axis=0*, *\*args*, *\*\*kwargs*)**

Agrega usando uma ou mais operações sobre o eixo especificado.

**.aggregate(*func=None*, *axis=0*, *\*args*, *\*\*kwargs*)**

Agrega usando uma ou mais operações sobre o eixo especificado.

**.transform(*func*, *axis=0*, *\*args*, *\*\*kwargs*)**

Chama ‘func on self’ produzindo um DataFrame com a mesma forma de eixo que self.

**.groupby(*by=None*, *axis=0*, *level=None*, *as\_index=True*, *sort=True*, *group\_keys=True*, *squeeze=NoDefault.no\_default*, *observed=False*, *dropna=True*)**

Agrupa o DataFrame usando um mapeador ou por uma série de colunas. Uma operação groupby envolve alguma combinação de dividir o objeto, aplicar uma função e combinar os resultados. Isso pode ser usado para agrupar grandes quantidades de dados e computar operações nesses grupos.

**MÉTODOS DE LIMPEZA e DADOS NULOS**

**.drop(*labels=None*, *axis=0*, *index=None*, *columns=None*, *level=None*, *inplace=False*, *errors='raise'*)**

Solta os rótulos especificados de linhas ou colunas.

**.drop\_duplicates(*subset=None*, *keep='first'*, *inplace=False*, *ignore\_index=False*)**

Retorna DataFrame com linhas duplicadas removidas. A consideração de certas colunas é opcional. Índices, incluindo índices de tempo, são ignorados.

**.duplicated(*subset=None*, *keep='first'*)**

Retorna boolean Series denotando linhas duplicadas. A consideração de certas colunas é opcional.

**.truncate(*before=None*, *after=None*, *axis=None*, *copy=True*)**

Trunque um Series ou DataFrame antes e depois de algum valor de índice. Essa é uma abreviação útil para indexação booleana com base em valores de índice acima ou abaixo de certos limites.

**.backfill(*axis=None*, *inplace=False*, *limit=None*, *downcast=None*)**

Sinônimo de DataFrame.fillna() com method='bfill'.

**.bfill(*axis=None*, *inplace=False*, *limit=None*, *downcast=None*)**

Sinônimo de DataFrame.fillna() com method='bfill'.

**.dropna(*axis=0*, *how='any'*, *thresh=None*, *subset=None*, *inplace=False*)**

Remova os valores ausentes.

**.ffill(*axis=None*, *inplace=False*, *limit=None*, *downcast=None*)**

Sinônimo de DataFrame.fillna() com method='ffill'.

**.fillna(*value=None*, *method=None*, *axis=None*, *inplace=False*, *limit=None*, *downcast=None*)**

Preencha os valores NA/NaN usando o método especificado.

**.interpolate(*method='linear'*, *axis=0*, *limit=None*, *inplace=False*, *limit\_direction=None*, *limit\_area=None*, *downcast=None*, *\*\*kwargs*)**

Preencha os valores NaN usando um método de interpolação.

**.isna()**

Detectar valores ausentes. Retorna um objeto booleano de mesmo tamanho indicando se os valores são NA. Valores NA, como None ou numpy.NaN, são mapeados para valores True. Todo o resto é mapeado para valores False. Caracteres como strings vazias '' ou numpy.inf não são considerados valores NA (a menos que você defina pandas.options.mode.use\_inf\_as\_na = True).

**.isnull()**

Detectar valores ausentes.

**.notna()**

Detectar valores existentes (não omissos).

**.notnull()**

Detectar valores existentes (não omissos).

**.pad(*axis=None*, *inplace=False*, *limit=None*, *downcast=None*)**

Sinônimo de DataFrame.fillna() com method='ffill'.

**MÉTODOS DE PLOTAGEM**

**.plot(*\*args*, *\*\*kwargs*)**

Faça plotagens de Series ou DataFrame.

**MÉTODOS SEM CLASSE**

**.xs(*key*, *axis=0*, *level=None*, *drop\_level=True*)**

Retorna uma seção cruzada da Series/Dataframe

**.clip(*lower=None*, *upper=None*, *axis=None*, *inplace=False*, *\*args*, *\*\*kwargs*)**

Valores de ajuste no(s) limite(s) de entrada. Atribui valores fora do limite aos valores do limite. Os limites podem ser valores singulares ou como arrays e, no último caso, o recorte é executado elemento a elemento no eixo especificado.

**.eval(*expr*, *inplace=False*, *\*\*kwargs*)**

Avalia uma string descrevendo operações em colunas DataFrame. Opera apenas em colunas, não em linhas ou elementos específicos. Isso permite que eval execute código arbitrário, o que pode torná-lo vulnerável à injeção de código se você passar a entrada do usuário para esta função.

**.at\_time(*time*, *asof=False*, *axis=None*)**

Seleciona valores em uma determinada hora do dia (por exemplo, 9h30).

**.between\_time(*start\_time*, *end\_time*, *include\_start=NoDefault.no\_default*, *include\_end=NoDefault.no\_default*, *inclusive=None*, *axis=None*)**

Seleciona valores entre horários específicos do dia (por exemplo, 9:00-9:30 AM).

**.equals(*other*)**

Testa se dois objetos contêm os mesmos elementos. Esta função permite que duas séries ou DataFrames sejam comparados entre si para ver se eles têm a mesma forma e elementos. NaNs no mesmo local são considerados iguais. O índice de linha/coluna não precisa ter o mesmo tipo, desde que os valores sejam considerados iguais. As colunas correspondentes devem ser do mesmo dtype.

**.nlargest(*n*, *columns*, *keep='first'*)**

Retorna as primeiras n linhas ordenadas por colunas em ordem decrescente. Retorna as primeiras n linhas com os maiores valores nas colunas, em ordem decrescente. As colunas que não são especificadas também são retornadas, mas não são usadas para ordenação.

**.nsmallest(*n*, *columns*, *keep='first'*)**

Retorna as primeiras n linhas ordenadas por colunas em ordem crescente. Retorna as primeiras n linhas com os menores valores nas colunas, em ordem crescente. As colunas que não são especificadas também são retornadas, mas não são usadas para ordenação.

**.swaplevel(*i=- 2*, *j=- 1*, *axis=0*)**

Troque os níveis i e j em um MultiIndex. O padrão é trocar os dois níveis mais internos do índice.

**.stack(*level=- 1*, *dropna=True*)**

Empilhe os níveis prescritos das colunas para o índice. Retorne um DataFrame ou Série remodelado com um índice de vários níveis com um ou mais novos níveis mais internos em comparação com o DataFrame atual. Os novos níveis mais internos são criados dinamizando as colunas do dataframe atual:

**.unstack(*level=- 1*, *fill\_value=None*)**

Dinamize um nível dos rótulos de índice (necessariamente hierárquicos). Retorna um DataFrame com um novo nível de rótulos de coluna cujo nível mais interno consiste nos rótulos de índice dinâmicos.

**.melt(*id\_vars=None*, *value\_vars=None*, *var\_name=None*, *value\_name='value'*, *col\_level=None*, *ignore\_index=True*)**

Despivote um DataFrame do formato amplo para o formato longo, opcionalmente deixando os identificadores definidos. Esta função é útil para massagear um DataFrame em um formato onde uma ou mais colunas são variáveis identificadoras (id\_vars), enquanto todas as outras colunas, consideradas variáveis medidas (value\_vars), são “unpivot” para o eixo da linha, deixando apenas duas não identificadoras colunas, 'variável' e 'valor'.