

Pandas:Series

Series: array unidimensional indexado que armazena valores de qualquer tipo.

`pd.Series(valores, index = array unidimensional)` - criar uma Series

`pd.read_excel(caminho, index_col= n, header=None).squeeze()` - criar uma Series a partir de um arquivo excel

<code>series.values</code>	Retorna os valores da series
<code>series.index</code>	Retorna os índices da series
<code>series.size</code>	Retorna a quantidade de registros da series
<code>series.head(n)</code>	Retorna as n primeiras linhas
<code>series.tail(n)</code>	Retorna as n últimas linhas

`series.loc[índice]` ou `series.loc[lista de índices]`
`series.iloc[posição]` ou `series.iloc[lista de posições]`

Retorna o valor do elemento indexado por *índice* ou uma nova Series com os elementos da *lista de índices*. `.loc` para os índices criados, `.iloc` para a posição no índice

`series.loc[índice]= valor` ou
`series.loc[lista de índices]= valor` ou `lista de valores`

Altera o valor/valores do elemento(s) indexado(s) por *índice/lista de índices*. Se o índice não existe, é incluído.

`series.drop(índice ou lista de índices)`
`series.dropna()`

drop: Retorna uma cópia da *series* sem os elementos da lista de índices.

dropna: Retorna uma cópia da *series* sem os elementos cujo valor é NaN

* com `inplace=True`, realiza a operação na Series, não cria uma cópia

`series.plot(kind='line', figsize=None, title=None ...)`

`kind= 'line'` : linha (default) `'bar'` : barra vertical `'barh'` : barra horizontal `'hist'` : histograma `'box'` : boxplot `'area'` : area plot
`'pie'` : pizza entre outros

`figsize = (altura,largura)` em polegadas

`title` = título do gráfico

`series.sort_values(ascending=True) *` – retorna uma cópia da Series ordenada pelos valores

* com `inplace=True`, realiza a operação na Series, não cria uma cópia

`series.sort_index(ascending=True) *` – retorna uma cópia da Series ordenada pelos labels dos índices

* com `inplace=True`, realiza a operação na Series, não cria uma cópia

`series.replace(to_replace=valor,value=novo) *` - Cria uma cópia, substituindo todas as ocorrências de valor por novo

* com `inplace=True`, realiza a operação na Series, não cria uma cópia

`series.update(series)` - Altera atuais valores pelos valores da Series recebida, alinhando pelo índice

`series.copy()` - Cria uma cópia dos dados e do index

`pd.concat(lista/dicionário de Series, ignore_index=False)` - Cria uma series com os elementos de todas as Series recebidas. Forma básica do concat.

`series.apply(função, args=(...))`

Aplica a função nos valores da Series, retornando uma nova Series. *Função* pode ser do Python ou definida pelo programador que opere sobre valores individuais da series.

`args = (...)` argumentos opcionais fornecidos à função

`series.fillna(value=None, ascending=True) *`

Retorna uma cópia do DataFrame substituindo valores Nan

value = scalar, dict, Series, or DataFrame

alternately a dict/Series/DataFrame of values specifying which value to use for each index (for a Series) or column (for a DataFrame). (values not in the dict/Series/DataFrame will not be filled).

axis = 0 ou 1

* com **inplace=True**, realiza a operação no DataFrame, não cria uma cópia

`series.unique()` – Retorna os valores exclusivos da Series

`series.nunique()` – Retornar o número de valores exclusivos na Series

`series.reindex(labels)` – Retorna uma cópia da Series na ordem especificada pela sequência de labels recebida. Não pode ser aplicado sobre índices com repetição

Sumarizações

Média: `series.mean()`

Mediana: `series.median()`

Moda: `series.mode()`

Máximo: `series.max()` [1]

Mínimo: `series.min()` [1]

Índice 1º Mínimo: `series.idxmin()`

Índice 1º Máximo: `series.idxmax()`

Quantil: `series.quantile(q=%)` padrão `q=0.5`

Variância: `series.var()`

Desvio Padrão: `series.std()`

Covariância: `series.cov(series)`

Correlação: `series.corr(series)`

Soma: `series.sum()`

Quantidade: `series.count()`

Contagem de valores exclusivos: `series.value_counts()` [1] (Tabela de frequências)

Resumo: `series.describe()`

[1] Operação aceita no atributo index

`pandas.cut(x, bins, right=True, labels=None, retbins=False, include_lowest=False)`

Retorna os índices das categorias/faixas (*bins*) de cada valor de *x*. (pode ser aplicado sobre o *index*)

x – array unidimensional a ser dividido em categorias (faixas) *localização* do arquivo: composto pelo caminho (absoluto/relativo) e nome

bins = int ou uma sequência de escalares.

Se bins é um int, define o número de categorias/faixas nas quais os valores de x serão divididos. Todas as faixas têm a mesma amplitude e o intervalo de x é estendido por 0,1% de cada lado para incluir os valores mínimo ou máximo de x.

Se bins é uma sequência, ela define os limites de cada categoria/faixa. Permite faixas de largura não uniforme. Nenhuma extensão do intervalo de x é feita.

right = True – indica se as faixas incluem o limite superior Ex. bins= [1,2,3,4] indicam (1,2), (2,3), (3,4).

labels = None ou array – se for especificado um array, este será usado como labels para as categorias/faixas resultantes.

retbins = False – se True, retorna uma tupla onde o segundo elemento é um array com os limites inferiores das faixas.

include_lowest = False – se True, o primeiro escalar da sequência é incluído no intervalo da primeira faixa.

series operador series

Objetos de Series semelhantes podem ser combinados com operações aritméticas. Os dados são alinhados pelo índice.

Retorna a Series resultante da operação. Se o índice € às duas Series, os valores são operados, nos demais casos retorna *Null/NaN*.

`series.metodoOperação(series, fill_value=valor)`

Objetos de Series semelhantes podem ser combinados com métodos que implementam as operações aritméticas. Os dados são alinhados pelo índice. Retorna a Series resultante da operação. Se o argumento `fill_value` está presente, quando não há sobreposição nos índices, utiliza *valor* para o cálculo. Quando ambas Series tem NaN no índice, retorna NaN.

valor in series

Retorna True se valor é um índice da Series ou False caso contrário.

series operador_lógico condição

Retorna uma nova Series de valores booleanos True/False.

- Pode-se usar a Series de booleanos para filtrar os itens selecionados (com valor True).

Conectivos Lógicos: e: & ou: | não: ~

series.isnull()

series.notnull()

Ambos métodos retornam uma nova Series com mesmos índices e:

isnull: True onde o valor está ausente (Null/NaN) e False, em caso contrário.

notnull: True onde o valor não está ausente e False, caso contrário

series.isin(lista de valores)

series.index.isin(lista de índices)

Retorna uma Series com True onde o elemento da Series € a lista de valores. Útil para selecionar linhas com os valores desejados.

Para o *Index* retorna um vetor booleano. Útil quando não se sabe quais dos rótulos procurados estão de fato presentes.

series.str.contains(padrão)

series.str.index.contains(padrão)

Retorna uma Series com True onde o elemento da Series € a lista de valores. Útil para selecionar linhas com os valores desejados.

Para o *Index* retorna um vetor booleano. Útil quando não se sabe quais dos rótulos procurados estão de fato presentes.

series.groupby(level=0)

Agrupa elementos com mesmo índice, atribuindo o valor do índice ao nome do grupo

Retorna um objeto GroupBy

GroupBy.groups

GroupBy.indices

GroupBy.groups dict {nome do grupo -> labels dos índices do grupo}

GroupBy.indices dict {nome do grupo -> array com a posição dos índices do grupo}

series.groupby(by=função)

Agrupa elementos com mesmo índice, atribuindo o valor do índice ao nome do grupo

Retorna um objeto GroupBy

GroupBy.get_group(nome)

Constrói uma Series com os elementos do grupo cujo nome foi fornecido

GroupBy.size()

Retorna a quantidade de elementos de cada grupo

GroupBy.agg(função pré-definida ou nome de método)

GroupBy.agg(lista de funções/nome de método)

As funções fornecidas para agregação reduzem a dimensão do objeto fornecido. São aplicadas sobre os valores do grupo e retornam um resultado para o conjunto. As mais comuns são *mean*, *sum*, *size*, *count*, *std*, *var*, *sem*, *describe*, *first*, *last*, *nth*, *min*, *max*.