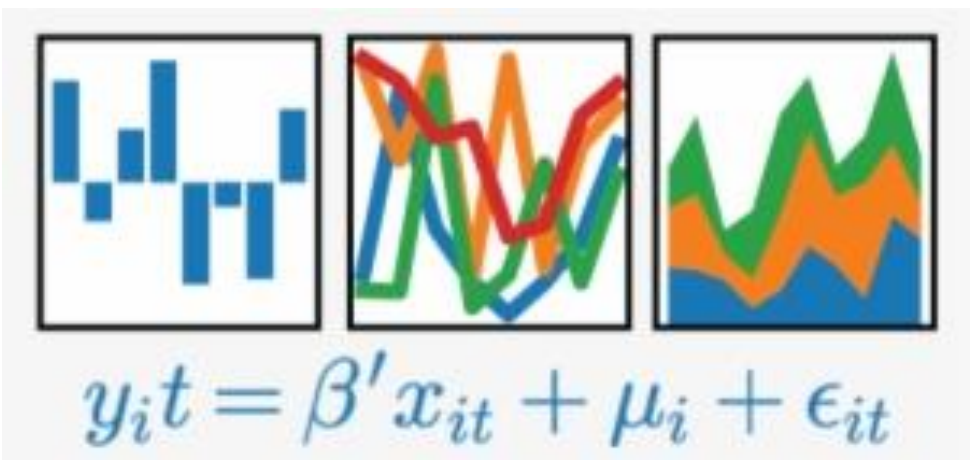
@Odemir Depieri Jr

Guia rápido sobre o framework Pandas



Criando uma serie de dados

Vamos importar a biblioteca pandas para podemos utilizar seus recursos.

```
[35] # Biblioteca para modelagem de dados
     import pandas as pd
```

A partir de agora os recursos do pandas foram apelidados de ‘pd’.

Vamos criar nossa serie de dados.

```
[49] # Lista com os rotulos
     Labels = ['1º', '2º', '3º']

     # Lista com os valores
     Valores = [10, 20, 30]

     # Criando a base de dados com as listas
     Serie = pd.Series( data=Valores, index=Labels )
     Serie
```

```
1º    10
2º    20
3º    30
dtype: int64
```

Definimos uma variável chamada ‘Serie’ e nela chamamos a função ‘pd.Series’ e passamos como parâmetros as listas que criamos.

Podemos selecionar apenas um rotulo da serie

```
[50] Serie['1º']
```

```
10
```

Podemos fazer operações matemáticas na serie

```
[51] Serie * 2
```

```
1º    20
2º    40
3º    60
dtype: int64
```

Nesse caso estamos aplicando um multiplicação na serie inteira.

Podemos criar outra serie e somar junto a serie anterior

```
[52] Serie_Nova = pd.Series( data=[50,100,150], index=Labels )

     Serie + Serie_Nova
```

```
1º     60
2º    120
3º    180
dtype: int64
```

Criando uma base de dados

Vamos criar uma base dados.

```
[68] # Criando um dicionário
     Dicionario = {
         'A' : [1, 2, 3],
         'B' : [-3, -2, -1],
         'C' : [0, 10, 20] }

     # Criando uma lista com os labels
     Label = ['1º Linha', '2º Linha', '3º Linha']

     DataFrame_01 = pd.DataFrame( Dicionario, index=Label )
     DataFrame_01
```

	A	B	C
1º Linha	1	-3	0
2º Linha	2	-2	10
3º Linha	3	-1	20

Selecionar apenas 1 (uma) coluna

```
[69] DataFrame_01['A']
```

```
1º Linha    1
2º Linha    2
3º Linha    3
Name: A, dtype: int64
```

Selecionar 2 colunas

```
[70] DataFrame_01[['A', 'B']]
```

	A	B
1º Linha	1	-3
2º Linha	2	-2
3º Linha	3	-1

Criando uma nova coluna e fazendo operações matemáticas entre as colunas

```
[71] DataFrame_01['Nova_Coluna'] = DataFrame_01['A'] * DataFrame_01['B']
     DataFrame_01
```

	A	B	C	Nova_Coluna
1º Linha	1	-3	0	-3
2º Linha	2	-2	10	-4
3º Linha	3	-1	20	-3

Para criar uma coluna basta nomeá-la e definir os parâmetros

Comandos sobre a base de dados

Excluir uma coluna

```
[73] DataFrame_01.drop('Nova_Coluna', axis=1, inplace=True)
     DataFrame_01
```

	A	B	C
1º Linha	1	-3	0
2º Linha	2	-2	10
3º Linha	3	-1	20

Nesse caso estamos excluindo uma coluna definitivamente da base de dados.

O parâmetro 'inplace=True' quer dizer que vamos excluir da base origem, caso fosse 'inplace=False' iria excluir apenas na situação atual.

O parâmetro 'axis=1' quer dizer que vamos excluir a coluna. Quando quiser excluir uma linha é usado 'axis=0'.

Localizar uma linha inteira

```
[80] DataFrame_01.loc['1º Linha']
```

```
A    1
B   -3
C    0
Name: 1º Linha, dtype: int64
```

Localizar diversas linhas e colunas

```
[81] DataFrame_01.loc[['1º Linha','3º Linha' ], ['A', 'C']]
```

	A	C
1º Linha	1	0
3º Linha	3	20

Localizar diversas linhas e colunas usando parâmetros numéricos através da posição da base de dados

```
[90] DataFrame_01.iloc[2:3, 1:]
```

	B	C
3º Linha	-1	20

```
DataFrame_01.iloc[2:3, 1:]
```

Linhas

Colunas

Comandos sobre a base de dados

Verificando se há valores menores de 0 na base inteira

```
[93] DataFrame_01 > 0
```

	A	B	C
1º Linha	True	False	False
2º Linha	True	False	True
3º Linha	True	False	True

Nesse caso estamos verificando em toda a base de dados se há valores maiores que zero.
Nesse contexto o pandas retorna se é verdadeiro ou false quanto a condição que passamos.

Filtrando dados na base de dados

```
[95] DataFrame_01[ DataFrame_01['A'] > 0 ]['C']
```

```
1º Linha     0
2º Linha    10
3º Linha    20
Name: C, dtype: int64
```

Nesse caso estamos :
1º Passando um parâmetro para verificar todos os casos maiores que 0 na coluna A
2º Retornando apenas os valores da coluna C

Filtrando dados na base de dados com parâmetros em variáveis

```
[96] Filtro = DataFrame_01['C'] > 0
     DataFrame_02 = DataFrame_01[Filtro]
     DataFrame_02['A']
```

```
2º Linha    2
3º Linha    3
Name: A, dtype: int64
```

Nesse caso estamos :
1º Criando uma variável na qual estamos passando como parâmetro todos os casos da coluna C maior que 0
    * Lembrando que nesse caso o pandas irá retornar verdadeiro ou falso
2º Criamos uma nova base de dados e passamos como filtro a variável definida na etapa 1
3º Retornando apenas os valores da coluna A

Comandos sobre a base de dados

Filtrando os dados com diversos parâmetros e condições

```
[97] DataFrame_01[
       ( DataFrame_01['A'] > 1 ) & ( DataFrame_01['C'] > 0 )
     ]
```

	A	B	C
2º Linha	2	-2	10
3º Linha	3	-1	20

Nesse caso estamos filtrando todos os casos da coluna A maior que 1 **e** passando outro parâmetro para filtrar todos os casos da coluna C maior 0.

Transformando o index em uma coluna

```
[102] DataFrame_01.reset_index()
```

	index	A	B	C
0	1º Linha	1	-3	0
1	2º Linha	2	-2	10
2	3º Linha	3	-1	20

Nesse caso transportamos o index das linhas para virar uma coluna na base de dados. Assim o index foi resetado e ficou como numérico a partir de agora.
Se colocar o comando 'inplace=True' entre os parênteses, esse comando será aplicado para a base de origem.

@Odemir Depieri Jr

Concatenar, Juntar e Mesclar

Criando as bases de dados para o exemplo

```
[109] # Criando varios dicionarios
      Dicionario_01 = {'A' : [1, 2, 3],
                       'B' : [-32, -21, -15],
                       'C' : [60, 10, 20],
                       'Chave' : ['AA', 'BB', 'CC'] }

      Dicionario_02 = {'A' : [6, 7, 8],
                       'B' : [-39, -28, -17],
                       'C' : [1000, 10, 60],
                       'Chave' : ['AA', 'BB', 'CC'] }

      Dicionario_03 = {'A' : [11, 12, 13],
                       'B' : [-39, -22, -11],
                       'C' : [30, 10, 20],
                       'Chave' : ['AA', 'BB', 'CC'] }

      # Criando varias listas para serem os labels
      Label_01 = ['1º Linha', '2º Linha', '3º Linha']
      Label_02 = ['4º Linha', '5º Linha', '6º Linha']
      Label_03 = ['7º Linha', '8º Linha', '9º Linha']

      # Estruturando as bases de dados
      DataFrame_01 = pd.DataFrame( Dicionario_01, index=Label_01 )
      DataFrame_02 = pd.DataFrame( Dicionario_02, index=Label_02 )
      DataFrame_03 = pd.DataFrame( Dicionario_03, index=Label_03 )
```

Empilhando os dados

```
[110] pd.concat(
        [ DataFrame_01, DataFrame_02, DataFrame_03 ]
      )
```

	A	B	C	Chave
1º Linha	1	-32	60	AA
2º Linha	2	-21	10	BB
3º Linha	3	-15	20	CC
4º Linha	6	-39	1000	AA
5º Linha	7	-28	10	BB
6º Linha	8	-17	60	CC
7º Linha	11	-39	30	AA
8º Linha	12	-22	10	BB
9º Linha	13	-11	20	CC

O comando **concat** irá empilhar todas as bases de dados, desde que todas tenham a mesma estrutura (colunas).

@Odemir Depieri Jr

Concatenar, Juntar e Mesclar

Função Mesclar permite que mescle os dados de diferentes base de dados. Essa função é semelhante a mesclagem de tabelas do SQL

```
[112] pd.merge(
          DataFrame_01, DataFrame_02, how='inner', on='Chave'
      )
```

	A_x	B_x	C_x	Chave	A_y	B_y	C_y
0	1	-32	60	AA	6	-39	1000
1	2	-21	10	BB	7	-28	10
2	3	-15	20	CC	8	-17	60

O comando **merge** uni as colunas baseando em uma chave, muito similar ao SQL.
O parâmetro 'how' há diversas forma:
- inner = apenas os casos que localizou em ambos os lados
- left = apenas os dados da base de dados do lado esquerdo
- right = apenas os dados da base de dados do lado direito
- outer = união das chaves em ambos os lados
- cross = cria o produto cartesiano de ambos os quadros

Função Juntar combina as colunas de ambas as bases de dados

```
# Criando varios dicionarios
Dicionario_01 = {'A' : [1, 2, 3],
                 'B' : [-32, -21, -15],
                 'C' : [60, 10, 20] }

Dicionario_02 = {'D' : [6, 7, 8],
                 'E' : [-39, -28, -17],
                 'F' : [1000, 10, 60] }

# Criando varias listas para serem os labels
Label_01 = ['1º Linha', '2º Linha', '3º Linha']


# Estruturando as bases de dados
DataFrame_01 = pd.DataFrame( Dicionario_01, index=Label_01 )
DataFrame_02 = pd.DataFrame( Dicionario_02, index=Label_01 )

# Aplicando a função join
DataFrame_01.join(DataFrame_02)
```

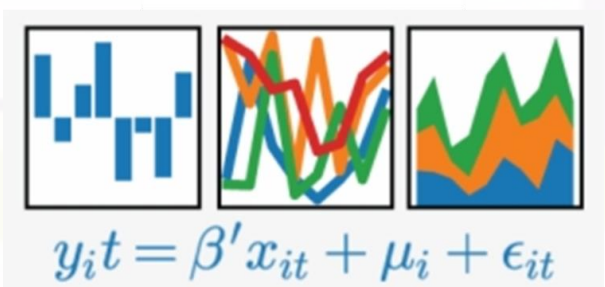
	A	B	C	D	E	F
1º Linha	1	-32	60	6	-39	1000
2º Linha	2	-21	10	7	-28	10
3º Linha	3	-15	20	8	-17	60

Colunas A,BC eram da base de dados 1 e as colunas D,E,F eram da base de dados 2. Assim o join uniu todas elas

Final

Esse guia é super rápido e apenas uma introdução sobre o tema.

Guia da documentação caso queira mais detalhes
https://pandas.pydata.org/docs/reference/io.html



Odemir Depieri Jr

Software Engineer Sr
Tech Lead
Specialization AI