



# Organizando Dataframes

- criando um dataframe do tipo lista de listas

```
data = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]  
data  
[[1, 2, 3], [4, 5, 6], [7, 8, 9]]
```

- pode-se criar uma lista de caractere para adicionar ao index e columns

```
#cria uma lista com cada caracter  
list('321')  
['3', '2', '1']
```

## Index e Columns em Dataframes

```
df = pd.DataFrame(data, list('321'), list('ZYX'))  
df  
   Z  Y  X  
3  1  2  3  
2  4  5  6  
1  7  8  9
```

## Organizando o index

```
# organiza o index e implementa a função dentro do Dataframe
df.sort_index(inplace=True)
df
   Z Y X
1 7 8 9
2 4 5 6
3 1 2 3
```

```
# organiza o index, grava a função no df e organiza as colunas
# o parâmetro axis=1 que organiza as colunas
df.sort_index(inplace=True, axis=1)
df
   X Y Z
1 9 8 7
2 6 5 4
3 3 2 1
```

## Organizando por valores

- função ⇒ sort\_values()
- essa função permite que seja escolhido o parâmetro by

```
# pega a coluna X como referência para colocar em ordem
df.sort_values(by='X', inplace=True)
df
   X Y Z
3 3 2 1
2 6 5 4
1 9 8 7
```

```
# organiza com base na linha 3
df.sort_values(by='3', axis=1, inplace=True)
df
   Z Y X
3 1 2 3
2 4 5 6
1 7 8 9
```

```

# também é possível passar uma lista como parâmetro
# se a lista for para linhas, deve acompanhar o parâmetro axis=1
df.sort_values(by=['X', 'Y'])
   Z  Y  X
3  1  2  3
2  4  5  6
1  7  8  9

```

## Exercícios

### 1. Considere o seguinte DataFrame nomeado como df:

Z=9,6,3

Y=8,5,2

X=7,4,1

Colunas = C,B,A

Após executar o seguinte conjunto de rotinas no Jupyter:

df.sort\_index()

df.sort\_index(axis = 1)

df

Qual o resultado obtido?

```

dados2 = [[9, 6, 3], [8, 5, 2], [7, 4, 1]]
df = pd.DataFrame(dados2, list('ZYX'), list('CBA'))
df.sort_index()
df.sort_index(axis = 1)
df
   C  B  A
Z  9  6  3
Y  8  5  2
X  7  4  1

```

- esse resultado foi gerado, por que o jupyter mostraria apenas o segundo comando, e também a função não foi gravada no dataframe com o parâmetro inplace=True, por isso quando é chamada a variável que remete ao dataframe, ele continua inalterado

## 2.

Observe o esquema abaixo:

The diagram illustrates a transformation process. On the left, there is a table labeled **df\_A**. On the right, there is a table labeled **df\_B**. A large grey arrow points from **df\_A** to **df\_B**, with the question "Código?" written inside the arrow.

	Nome	Sexo	Idade
0	Ary	M	21
1	Cátia	F	19
2	Carlos	M	50
3	Beto	M	29
4	Bruna	F	31
5	Ana	F	42

	Nome	Sexo	Idade
5	Ana	F	42
4	Bruna	F	31
1	Cátia	F	19
0	Ary	M	21
3	Beto	M	29
2	Carlos	M	50

Qual das linhas de código abaixo transforma **df\_A** em **df\_B**?

```
df_A = pd.DataFrame([['Ary', 'M', 21], ['Cátia', 'F', 19],
                     ['Carlos', 'M', 50], ['Beto', 'M', 29],
                     ['Bruna', 'F', 31], ['Ana', 'F', 42]],
                     columns=['Nome', 'Sexo', 'Idade'])
```

```
df_A
    Nome  Sexo  Idade
0 Ary      M    21
1 Cátia    F    19
2 Carlos   M    50
3 Beto     M    29
4 Bruna    F    31
5 Ana      F    42
```

```
df_B = df_A.sort_values(by=['Sexo', 'Nome'])
```

```
df_B
    Nome  Sexo  Idade
5 Ana      F    42
4 Bruna   F    31
1 Cátia   F    19
0 Ary     M    21
3 Beto    M    29
2 Carlos  M    50
```

- 
- se chamar primeiro a variável 'Nome' ele irá gerar outro resultado
  - será organizado apenas os nomes por ordem alfabética e não será organizado o sexo
  - deve-se tomar esse tipo de cuidado ao organizar um dataframe