

Lista6A

December 23, 2021

ESTATÍSTICA APLICADA

ENGENHARIA DE COMPUTAÇÃO

LISTA 6A

ALUNO: Maria Eduarda Pereira de Souza Melo

QUESTÃO1: Junção no sentido da linha

```
[1]: import pandas as pd
Mercearia = pd.DataFrame({"loja":["A", "A", "A"],
"dia": ["sex", "sab", "dom"],
"valor": [7500, 9500, 8200]})

Farmácia = pd.DataFrame({"loja":["B", "B", "B"],
"dia": ["sex", "sab", "dom"],
"valor": [5100, 8250, 9900]})

Restaurante = pd.DataFrame({"loja":["C", "C"],
"dia": ["sab", "dom"],
"valor": [7500, 11800]})
x = pd.concat([Mercearia, Farmácia,Restaurante])
x
```

```
[1]:   loja  dia  valor
0    A  sex   7500
1    A  sab   9500
2    A  dom   8200
0    B  sex   5100
1    B  sab   8250
2    B  dom   9900
0    C  sab   7500
1    C  dom  11800
```

QUESTÃO 1: No sentido das colunas

```
[2]: y = pd.concat([Mercearia, Farmácia,Restaurante], axis=1)
y
```

```
[2]:   loja  dia  valor loja  dia  valor loja  dia  valor
      0   A  sex   7500   B  sex   5100   C  sab   7500.0
      1   A  sab   9500   B  sab   8250   C  dom  11800.0
      2   A  dom   8200   B  dom   9900  NaN   NaN     NaN
```

Por padrão o concat adiciona os dataframes no sentido de linha (axis = 0), então quando especificamos o axis=1 estamos fazendo a junção por colunas. O valor NaN apareceu pois o dataframe restaurante possui apenas dois índices.

QUESTÃO 2A:

```
[3]: SQL = pd.DataFrame({"email": ["rakesh@xyz.com",
                                   "ecg@acmecorpus.com"]})

PYTHON = pd.DataFrame({"email": ["ana@xyz.com",
                                   "jonas@acmecorpus.com",
                                   "rakesh@xyz.com"]})

iguais = pd.concat([SQL, PYTHON])
x = iguais.drop_duplicates('email', keep=False)
x
```

```
[3]:           email
1    ecg@acmecorpus.com
0         ana@xyz.com
1  jonas@acmecorpus.com
```

QUESTÃO 2B:

```
[4]: duplicados = pd.merge(SQL, PYTHON)
      duplicados
```

```
[4]:           email
0  rakesh@xyz.com
```

QUESTÃO 2C: Email do indivíduo que fez apenas o curso de SQL

```
[5]: Repetidos = pd.concat([SQL, duplicados])
      unico = Repetidos.drop_duplicates('email', keep=False)
      unico
```

```
[5]:           email
1  ecg@acmecorpus.com
```

Email dos indivíduos que fizeram apenas o curso de PYTHON

```
[6]: Repetidos = pd.concat([PYTHON, duplicados])
      unico = Repetidos.drop_duplicates('email', keep=False)
      unico
```

```
[6]:          email
0          ana@xyz.com
1  jonas@acmecorpus.com
```

QUESTÃO 3:

```
[7]: dic_emp = {"id":["D1","D2","D3","D4"],
"nomDepto": ["Compras","RH","TI","Vendas"],
"local":["SP","RJ","RJ","SP"]}

dic_func = {"num":[3199,3269,3555,3788,3844],
"nome": ["Ana","David","José","Marina","Luís"],
"salario": [1600,2975,1500,5000,3000],
"idDepto": ["D2","D3",None,"D2","D4"]}

empresa = pd.DataFrame(dic_emp)
funcionarios = pd.DataFrame(dic_func)
NOVO = pd.merge(funcionarios,empresa,how="left",left_on="idDepto",right_on="id")
NOVO
```

```
[7]:   num  nome  salario idDepto   id nomDepto local
0  3199   Ana    1600      D2    D2      RH    RJ
1  3269  David    2975      D3    D3      TI    RJ
2  3555   José    1500    None   NaN      NaN   NaN
3  3788  Marina    5000      D2    D2      RH    RJ
4  3844   Luís    3000      D4    D4  Vendas    SP
```