

Python - Pandas



Pandas

- O pandas permite trabalhar com diferentes tipos de dados, por exemplo:
- Dados tabulares, como uma planilha Excel ou uma tabela SQL;
- Dados ordenados de modo temporal ou não;
- Matrizes;
- Qualquer outro conjunto de dados, que não necessariamente precisem estar rotulados;
- Biblioteca muito popular por ter a facilidade de ler, manipular, agregar e exibir os dados com poucos comandos.

Pandas – Estruturas de Dados

Principais objetos:

DataFrame:

- uma estrutura de dados tabular, orientada a colunas, com rótulos (labels) tanto para linhas quanto para colunas

Series:

- Uma Serie é uma matriz unidimensional que contém uma sequência de valores que apresentam uma indexação (que podem ser numéricos inteiros ou rótulos), muito parecida com uma única coluna no Excel.

Pandas

Series

		nome			idade
Índice	0	Maria	0	25	Linhas
	1	José	1	56	
	2	Ana	2	31	
	3	Paulo	3	43	

DataFrame

	Colunas	
	nomes	idade
0	Maria	25
1	José	56
2	Ana	31
3	Paulo	43

Pandas

Principais vantagens:

- A facilidade de aprender e de utilizar a biblioteca
- Comunidade crescente e muito ativa
- Suporte para alinhamento automático ou explícito dos dados
- Tratamento flexível e simplificado de dados ausentes
- Combinações e operações relacionais
- Informações estatísticas
- Séries temporais
- Facilidade para que usuários criem visualizações simplificadas de dados
- Vasta documentação

Pandas

Instalando o Pandas Python

```
PS C:\Users\DELL> pip install pandas
```

```
Collecting pandas
```

```
  Downloading pandas-2.0.0-cp39-cp39-win_amd64.whl (11.3 MB)
```

```
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 11.3/11.3 MB 17.2 MB/s eta 0:00:00
```

```
Collecting pytz>=2020.1
```

```
  Downloading pytz-2023.3-py2.py3-none-any.whl (502 kB)
```

```
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 502.3/502.3 kB 15.9 MB/s eta 0:00:00
```

```
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\dell\appdata\local\packages\pythonsoftwarefoundation.python.3.9_qbz5n2kfra8p0\localcache\local-packages\python39\site-packages (from pandas) (2.8.2)
```

```
Requirement already satisfied: numpy>=1.20.3 in c:\users\dell\appdata\local\packages\pythonsoftwarefoundation.python.3.9_qbz5n2kfra8p0\localcache\local-packages\python39\site-packages (from pandas) (1.24.2)
```

```
Collecting tzdata>=2022.1
```

```
  Downloading tzdata-2023.3-py2.py3-none-any.whl (341 kB)
```

```
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 341.8/341.8 kB 10.7 MB/s eta 0:00:00
```

```
Requirement already satisfied: six>=1.5 in c:\users\dell\appdata\local\packages\pythonsoftwarefoundation.python.3.9_qbz5n2kfra8p0\localcache\local-packages\python39\site-packages (from python-dateutil>=2.8.2->pandas) (1.16.0)
```

```
Installing collected packages: pytz, tzdata, pandas
```

```
Successfully installed pandas-2.0.0 pytz-2023.3 tzdata-2023.3
```

Pandas

Por onde começar?

Google Colab



Tec_emergentes_2023-1.ipynb ☆

Arquivo Editar Ver Inserir Ambiente de execução Ferramentas Ajuda



+ Código + Texto



✓
1s



```
1 import pandas as pd
```

{x}



Pandas

Leitura de um conjunto de dados (Dataset)

No Pandas existem vários métodos para leitura de dados com diferentes formatos (como .xlsx, json, .csv). Geralmente, esses métodos iniciam com a palavra *'read_'* seguido da extensão do arquivo.

```
df = pd.read_csv(url, sep = ';')  
type(df)
```

```
pandas.core.frame.DataFrame
```


Pandas

Visualizando um conjunto de linhas do Dataframe

Método: head



```
1 import pandas as pd
2 url = 'https://raw.githubusercontent.com/brunamulinari/Ba
3 df = pd.read_csv(url, sep = ';')
4 type(df)
5 df.head(n=6)
```



	id	data_aq	produto	quantidade	valor UN	Total	setor
0	0	01/01/2019	toalha	6	R\$ 35,00	R\$ 210,00	Mesa_banho
1	1	02/01/2019	toalha	6	R\$ 35,00	R\$ 210,00	--
2	2	03/01/2019	toalha	2	R\$ 35,00	R\$ 70,00	mesa_banho
3	3	01/02/2019	toalha	5	R\$ 35,00	R\$ 175,00	nada
4	4	02/02/2019	toalha	9	R\$ 35,00	R\$ 315,00	mesa_banho
5	5	04/01/2019	NaN	2	R\$ 35,00	R\$ 70,00	mesa_banho



Pandas

Visualizando um conjunto de linhas do Dataframe

pode utilizar de um recurso do Pandas que ao carregar esse conjunto de dados indique quais valores ou mensagens também devem ser considerados **ausentes**, **NaN**. Esse recuso é um parâmetro na função de 'read_' chamado *na_values*

```
df = pd.read_csv('https://raw.githubusercontent.com/brunamulinari/BasicPython  
Projects/main/Base_ficticia/baseficticia.csv', sep = ';', na_values=['--',  
'n/a', 'nada'])
```

Pandas

```
7 df.head(n=6)
```



	id	data_aq	produto	quantidade	valor UN	Total	setor
0	0	01/01/2019	toalha	6	R\$ 35,00	R\$ 210,00	Mesa_banho
1	1	02/01/2019	toalha	6	R\$ 35,00	R\$ 210,00	NaN
2	2	03/01/2019	toalha	2	R\$ 35,00	R\$ 70,00	mesa_banho
3	3	01/02/2019	toalha	5	R\$ 35,00	R\$ 175,00	NaN
4	4	02/02/2019	toalha	9	R\$ 35,00	R\$ 315,00	mesa_banho
5	5	04/01/2019	NaN	2	R\$ 35,00	R\$ 70,00	mesa_banho

Pandas

Visualizando as n últimas linhas do conjunto

Método: tail

```
8 df.tail(n=5)
```

```
9
```



	id	data_aq	produto	quantidade	valor UN	Total	setor
544	544	31/07/2019	quebra_cabeca	9	R\$ 19,99	R\$ 179,91	brinquedos
545	545	01/08/2019	quebra_cabeca	3	R\$ 19,99	R\$ 59,97	brinquedos
546	546	30/07/2019	quebra_cabeca	4	R\$ 19,99	R\$ 79,96	brinquedos
547	547	31/07/2019	quebra_cabeca	1	R\$ 19,99	R\$ 19,99	brinquedos
548	548	2019-03-03 00:00:00	quebra_cabeca	8	R\$ 19,99	R\$ 159,92	Brinquedos



Pandas

Descobrimos quantas informações esse conjunto de dados apresenta.

Comando: shape

```
9 df.shape
```

```
10
```

```
(549, 7)
```

Pandas

Já para saber que formato se encontram os dados em cada coluna, além da quantidade de memória para ler esse conjunto de dados, podemos utilizar o comando ***info***:

```
10 df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 549 entries, 0 to 548
Data columns (total 7 columns):
 #   Column      Non-Null Count  Dtype  
---  -
 0   id          549 non-null   int64  
 1   data_aq     549 non-null   object
 2   produto    542 non-null   object  
 3   quantidade  549 non-null   int64  
 4   valor UN   549 non-null   object  
 5   Total       549 non-null   object  
 6   setor       535 non-null   object  
dtypes: int64(2), object(5)
memory usage: 30.1+ KB
```

Em geral, quando a biblioteca não consegue identificar o tipo do dado entre os padrões python conhecidos (*int*, *float*, *string*, *datetime*, entre outros), ela define o dado com o formato de *object*.

Pandas

Visualizar quais são nossas colunas existentes e até mesmo alterar esses nomes, basta passar o novo conjunto de nomes desejados com a mesma quantidade de colunas existente no conjunto original:

```
11 df.columns
12 df.columns = ['id', 'data_aq', 'produto', 'quantidade', 'valor_un', 'valor_total', 'setor']
13 df.columns
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 549 entries, 0 to 548
Data columns (total 7 columns):
#   Column      Non-Null Count  Dtype
---  -
0   id           549 non-null    int64
1   data_aq      549 non-null    object
2   produto      542 non-null    object
3   quantidade   549 non-null    int64
4   valor UN     549 non-null    object
5   Total        549 non-null    object
6   setor        535 non-null    object
dtypes: int64(2), object(5)
memory usage: 30.1+ KB
Index(['id', 'data_aq', 'produto', 'quantidade', 'valor_un', 'valor_total',
      'setor'],
      dtype='object')
```

Pandas

Para verificar quantos dados faltantes existem em nosso conjunto, podemos utilizar a função ***isnull***, na qual verifica em cada uma das colunas se o elemento é nulo ou não, seguida da função ***sum***, que irá somar todas as respostas verdadeiras obtidas na função anterior

```
df.isnull().sum()
```

```
>>> <class 'pandas.core.frame.DataFrame'>
RangeIndex: 549 entries, 0 to 548
Data columns (total 7 columns):
#   Column      Non-Null Count  Dtype
---  -
0   id           549 non-null   int64
1   data_aq      549 non-null   object
2   produto      542 non-null   object
3   quantidade   549 non-null   int64
4   valor_UN     549 non-null   object
5   Total        549 non-null   object
6   setor        535 non-null   object
dtypes: int64(2), object(5)
memory usage: 30.1+ KB
id           0
data_aq      0
produto      7
quantidade   0
valor_un     0
valor_total  0
setor       14
dtype: int64
```