

# Módulo pandas

Catarina Oliveira

DCT DEPARTAMENTO CIÊNCIA  
E TECNOLOGIA

## CONTEÚDO

1. O que é
2. Por que utilizar
3. O que permite fazer
4. Exemplo
5. Exemplo com ficheiro CSV
6. Exemplo com ficheiro JSON
7. Acesso a dados em DataFrames
8. Ver dados do DataFrame
9. Acesso a parte dos dados em DataFrames: loc
10. Acesso a parte dos dados em DataFrames: iloc
11. Modificar dados de um DataFrame
12. Limpeza de dados: preenchimento de todos os Null e NaN
13. Limpeza de dados
14. Medidas estatísticas de colunas ou linhas
15. Operações com colunas ou linhas
16. Análise estatística
17. Gráficos

## O que é

- Módulo usado para análise de dados
- Permite:
  - Manipular estruturas de dados de dois tipos:
    - Séries
    - DataFrames
  - Atribuir nomes a linhas/colunas
  - Operações sobre dados:
    - Analisar
    - Limpar
    - Explorar
    - Manipular e transformar
- Tem suporte para dados em falta

Série

A	B	C	D	A
10	50	23	70	34

DataFrame

Índice	Age	Gender	Rating
Steve	32	Male	3.45
Lia	28	Female	4.6
Steve	45	Male	3.9
Katie	38	Female	2.78

Documentação: <https://pandas.pydata.org/docs/>

## Por que utilizar

- Permite analisar big data e tirar conclusões com base em teorias estatísticas.
- Tem mecanismos para limpar conjuntos de dados confusos e torná-los legíveis e relevantes.
- Dados relevantes são muito importantes em ciência de dados.

## O que permite fazer

### **Limpeza de dados:**

- Pode excluir linhas que não são relevantes ou contêm valores incorretos, como valores vazios ou NULL.

### **Análise de dados:**

- Fornece respostas sobre os dados como, por exemplo:
  - Existe uma correlação entre duas (ou mais) colunas?
  - Qual é o valor médio?
  - Valor máximo?
  - Valor mínimo?


## Exemplo

```
import pandas as pd

mydataset = {
    'cars': ["BMW", "Volvo", "Ford"],
    'passings': [3, 7, 2]
}

myvar = pd.DataFrame(mydataset)

print(myvar)
```



	cars	passings
0	BMW	3
1	Volvo	7
2	Ford	2

## Exemplo com ficheiro CSV

dados.csv

1	CodPostal,Cidade,Freguesia,Morada,Instituição
2	4200,Porto,Paranhos,Dr. António Bernardino de Almeida,Universidade Portucalense
3	4000,Porto,Santo Ildefonso,Praça do General Humberto Delgado,Câmara Municipal do Porto

```
import pandas as pd  
  
df = pd.read_csv("dados.csv")  
print(df)
```

	CodPostal	Cidade	Freguesia	Morada	Instituição
0	4200	Porto	Paranhos	Dr. António Bernardino de Almeida	Universidade Portucalense
1	4000	Porto	Santo Ildefonso	Praça do General Humberto Delgado	Câmara Municipal do Porto

## Exemplo com ficheiro JSON

```
import pandas as pd

df = pd.read_json("dados.json")
print(df)
```

```
dados.json
1 {
2   "instituições": [
3     {
4       "CodPostal": 4200,
5       "Cidade": "Porto",
6       "Freguesia": "Paranhos",
7       "Morada": "Rua Dr. António Bernardino de Almeida",
8       "Instituição": "Universidade Portucalense"
9     },
10    {
11      "CodPostal": 4000,
12      "Cidade": "Porto",
13      "Freguesia": "Santo Ildefonso",
14      "Morada": "Praça do General Humberto Delgado",
15      "Instituição": "Câmara Municipal do Porto"
16    }
17  ]
18 }
```

```
instituições
0 {'CodPostal': 4200, 'Cidade': 'Porto', 'Freguesia': 'Paranhos', 'Morada': 'Dr. António Bernardino de Almeida', 'Instituição': 'Universidade Portucalense'}
1 {'CodPostal': 4000, 'Cidade': 'Porto', 'Freguesia': 'Santo Ildefonso', 'Morada': 'Praça do General Humberto Delgado', 'Instituição': 'Câmara Municipal do Porto'}
```



## Acesso a dados em DataFrames

inventario.csv	
1	Produto, Preço, Quantidade
2	Café, 1.3, 4300
3	Águas, 0.21, 8000
4	Leite, , 6000
5	Chocolate, 0.35,
6	Café, 1.25, 3200
7	Leite, , 9500
8	Chocolate, 0.36,
9	Café, 1.3, 2900

Mostrar todo o DataFrame

```
import pandas as pd

df = pd.read_csv("inventario.csv")
print(df)
```

	Produto	Preço	Quantidade
0	Café	1.30	4300.0
1	Águas	0.21	8000.0
2	Leite	NaN	6000.0
3	Chocolate	0.35	NaN
4	Café	1.25	3200.0
5	Leite	NaN	9500.0
6	Chocolate	0.36	NaN
7	Café	1.30	2900.0

Mostrar só a coluna "Preço"

```
import pandas as pd

df = pd.read_csv("inventario.csv")
print(df['Preço'])
```

```
0    1.30
1    0.21
2    NaN
3    0.35
4    1.25
5    NaN
6    0.36
1.30
Name: Preço, dtype: float64
```

## Ver dados do DataFrame

Mostrar as 3 primeiras linhas

```
import pandas as pd

df = pd.read_csv("inventario.csv")
print(df.head(3))
```

	Produto	Preço	Quantidade
0	Café	1.30	4300.0
1	Águas	0.21	8000.0
2	Leite	NaN	6000.0

Mostrar as 3 últimas linhas

```
import pandas as pd

df = pd.read_csv("inventario.csv")
print(df.tail(3))
```

	Produto	Preço	Quantidade
5	Leite	NaN	9500.0
6	Chocolate	0.36	NaN
7	Café	1.30	2900.0

Mostrar informação

```
import pandas as pd

df = pd.read_csv("inventario.csv")
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8 entries, 0 to 7
Data columns (total 3 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   Produto     8 non-null      object
1   Preço       6 non-null      float64
2   Quantidade  6 non-null      float64
dtypes: float64(2), object(1)
memory usage: 320.0+ bytes
```

## Acesso a parte dos dados em DataFrames: loc

loc: permite aceder a um grupo de linhas e colunas a partir dos labels ou de um vetor de booleanos

<https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.loc.html>

```
import pandas as pd

df = pd.read_csv("inventario.csv")
print(df['Quantidade'] >= 5000)
```

Que linhas têm “Quantidade” superior a 5000

```
0    False
1     True
2     True
3    False
4    False
5     True
6    False
7    False
Name: Quantidade, dtype: bool
```

```
import pandas as pd

df = pd.read_csv("inventario.csv")
print(df.loc[df['Quantidade'] >= 5000])
```

Mostrar as linhas que têm “Quantidade” superior a 5000

	Produto	Preço	Quantidade
1	Águas	0.21	8000.0
2	Leite	NaN	6000.0
5	Leite	NaN	9500.0

```
import pandas as pd

df = pd.read_csv("inventario.csv")
print(df.loc[df['Quantidade'] >= 5000, 'Preço'])
```

Mostrar a coluna “Preço” das linhas que têm “Quantidade” superior a 5000

```
1    0.21
2    NaN
5    NaN
Name: Preço, dtype: float64
```

## Acesso a parte dos dados em DataFrames: iloc

iloc: permite aceder a um grupo de linhas e colunas a partir dos seus índices

<https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.iloc.html>

```
import pandas as pd

df = pd.read_csv("inventario.csv")
print(df.iloc[0:2, :])
```

Mostrar as linhas com índices 0 a 2 (0, 1, 2) de todas as colunas

```
import pandas as pd

df = pd.read_csv("inventario.csv")
print(df.iloc[:, 0:2])
```

Mostrar todas as linhas das colunas com índices 0 a 2

```
import pandas as pd

df = pd.read_csv("inventario.csv")
print(df.iloc[0:2, 0:2])
```

Mostrar linhas com índices 0 a 2 das colunas com índices 0 a 2

	Produto	Preço	Quantidade
0	Café	1.30	4300.0
1	Águas	0.21	8000.0

	Produto	Preço
0	Café	1.30
1	Águas	0.21
2	Leite	NaN
3	Chocolate	0.35
4	Café	1.25
5	Leite	NaN
6	Chocolate	0.36
7	Café	1.30


	Produto	Preço
0	Café	1.30
1	Águas	0.21

## Modificar dados de um DataFrame

Alterar todos os preços para 1.5

```
import pandas as pd

df = pd.read_csv("inventario.csv")
df['Preço'] = 1.5
print(df)
```

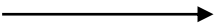


	Produto	Preço	Quantidade
0	Café	1.5	4300.0
1	Águas	1.5	8000.0
2	Leite	1.5	6000.0
3	Chocolate	1.5	NaN
4	Café	1.5	3200.0
5	Leite	1.5	9500.0
6	Chocolate	1.5	NaN
7	Café	1.5	2900.0

Alterar todos os preços dos produtos com quantidade superior a 5000 para 1.5

```
import pandas as pd

df = pd.read_csv("inventario.csv")
df.loc[df['Quantidade'] >= 5000, 'Preço'] = 1.5
print(df)
```



	Produto	Preço	Quantidade
0	Café	1.30	4300.0
1	Águas	1.50	8000.0
2	Leite	1.50	6000.0
3	Chocolate	0.35	NaN
4	Café	1.25	3200.0
5	Leite	1.50	9500.0
6	Chocolate	0.36	NaN
7	Café	1.30	2900.0

## Limpeza de dados: preenchimento de todos os Null e NaN

Substituir no DataFrame original

```
import pandas as pd

df = pd.read_csv("inventario.csv")
df.fillna(888, inplace = True)
print(df)
```

	Produto	Preço	Quantidade
0	Café	1.30	4300.0
1	Águas	0.21	8000.0
2	Leite	NaN	6000.0
3	Chocolate	0.35	NaN
4	Café	1.25	3200.0
5	Leite	NaN	9500.0
6	Chocolate	0.36	NaN
	Café	1.30	2900.0

	Produto	Preço	Quantidade
0	Café	1.30	4300.0
1	Águas	0.21	8000.0
2	Leite	888.00	6000.0
3	Chocolate	0.35	888.0
4	Café	1.25	3200.0
5	Leite	888.00	9500.0
6	Chocolate	0.36	888.0
7	Café	1.30	2900.0

	Produto	Preço	Quantidade
0	Café	1.30	4300.0
1	Águas	0.21	8000.0
2	Leite	888.00	6000.0
3	Chocolate	0.35	888.0
4	Café	1.25	3200.0
5	Leite	888.00	9500.0
6	Chocolate	0.36	888.0
7	Café	1.30	2900.0

Criar um novo DataFrame

```
import pandas as pd

df = pd.read_csv("inventario.csv")
novoDF = df.fillna(888)

print(df)
print(novoDF)
```


## Limpeza de dados

Substituir no DataFrame original, apenas na coluna Quantidade

```
import pandas as pd

df = pd.read_csv("inventario.csv")
df["Quantidade"].fillna(888, inplace = True)

print(df)
```




	Produto	Preço	Quantidade
0	Café	1.30	4300.0
1	Águas	0.21	8000.0
2	Leite	NaN	6000.0
3	Chocolate	0.35	888.0
4	Café	1.25	3200.0
5	Leite	NaN	9500.0
6	Chocolate	0.36	888.0
7	Café	1.30	2900.0

Criar um novo DataFrame (df não é alterado) sem as linhas com NaN

```
import pandas as pd

df = pd.read_csv("inventario.csv")
novoDF = df.dropna( )

print(novoDF)
```




	Produto	Preço	Quantidade
0	Café	1.30	4300.0
1	Águas	0.21	8000.0
4	Café	1.25	3200.0
7	Café	1.30	2900.0

Remover as linhas com NaN do DataFrame original

```
import pandas as pd

df = pd.read_csv("inventario.csv")
df.dropna(inplace = True)

print(df)
```



	Produto	Preço	Quantidade
0	Café	1.30	4300.0
1	Águas	0.21	8000.0
4	Café	1.25	3200.0
7	Café	1.30	2900.0

## Medidas estatísticas de colunas ou linhas

- **Mean** (média)
- **Median** (mediana)
- **Max** (máximo)
- **Min** (mínimo)
- **Std** (desvio padrão)

Obter a média da coluna “Preço” e a média dos valores numéricos da linha 1 (índices das linhas começam em 0)

```
import pandas as pd

df = pd.read_csv("inventario.csv")

print(df['Preço'].mean())
print(df.iloc[1,1:3].mean())
```

0.7949999999999999  
4000.105



## Operações com colunas ou linhas

- Soma
- Subtração
- Divisão
- ...

Obter o produto dos preços pela quantidades

```
import pandas as pd

df = pd.read_csv("inventario.csv")

print(df['Preço'] * df['Quantidade'])
```

```
0    5590.0
1    1680.0
2         NaN
3         NaN
4    4000.0
5         NaN
6         NaN
7    3770.0
dtype: float64
```

Obter a soma dos preços e quantidades das linhas 0 e 1

```
import pandas as pd

df = pd.read_csv("inventario.csv")

print(df.iloc[0, 1:3] + df.iloc[1, 1:3])
```

```
Preço    1.51
Quantidade 12300.0
dtype: object
```

## Análise estatística

```
import pandas as pd

df = pd.read_csv("inventario.csv")

print(df.describe())
```

	Preço	Quantidade
count	6.000000	6.000000
mean	0.795000	5650.000000
std	0.537875	2677.87229
min	0.210000	2900.000000
25%	0.352500	3475.000000
50%	0.805000	5150.000000
75%	1.287500	7500.000000
max	1.300000	9500.000000

```
import pandas as pd

df = pd.read_csv("inventario.csv")

print(df.iloc[:, 1:3].corr())
```

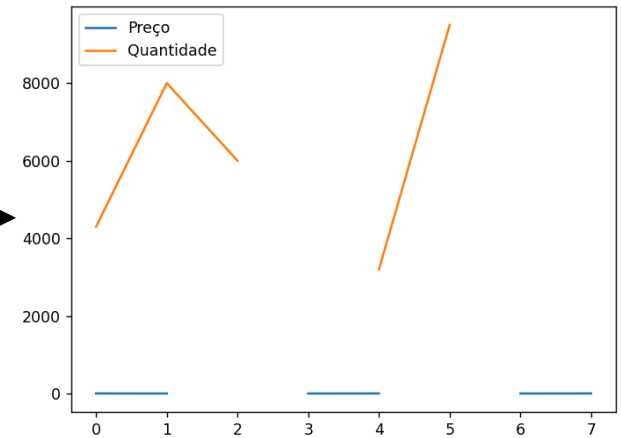
	Preço	Quantidade
Preço	1.000000	-0.962051
Quantidade	-0.962051	1.000000

## Gráficos

```
import pandas as pd
from matplotlib import pyplot as plt

df = pd.read_csv("inventario.csv")

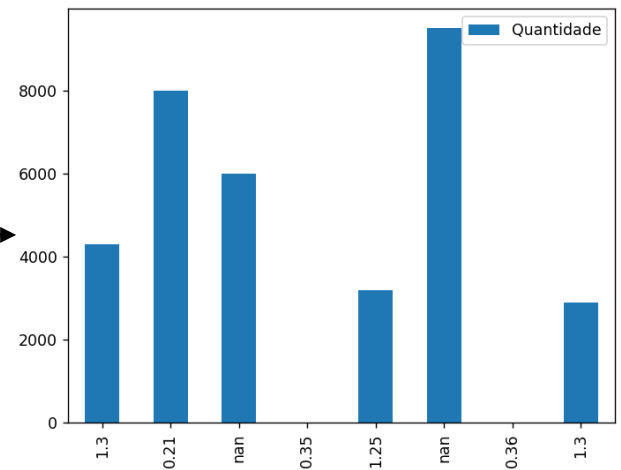
df.plot()
plt.show()
```



```
import pandas as pd
from matplotlib import pyplot as plt

df = pd.read_csv("inventario.csv")

df.plot(x = 'Preço', y = 'Quantidade', kind='bar')
plt.show()
```





UNIVERSIDADE  
PORTUCALENSE

Do conhecimento à prática.