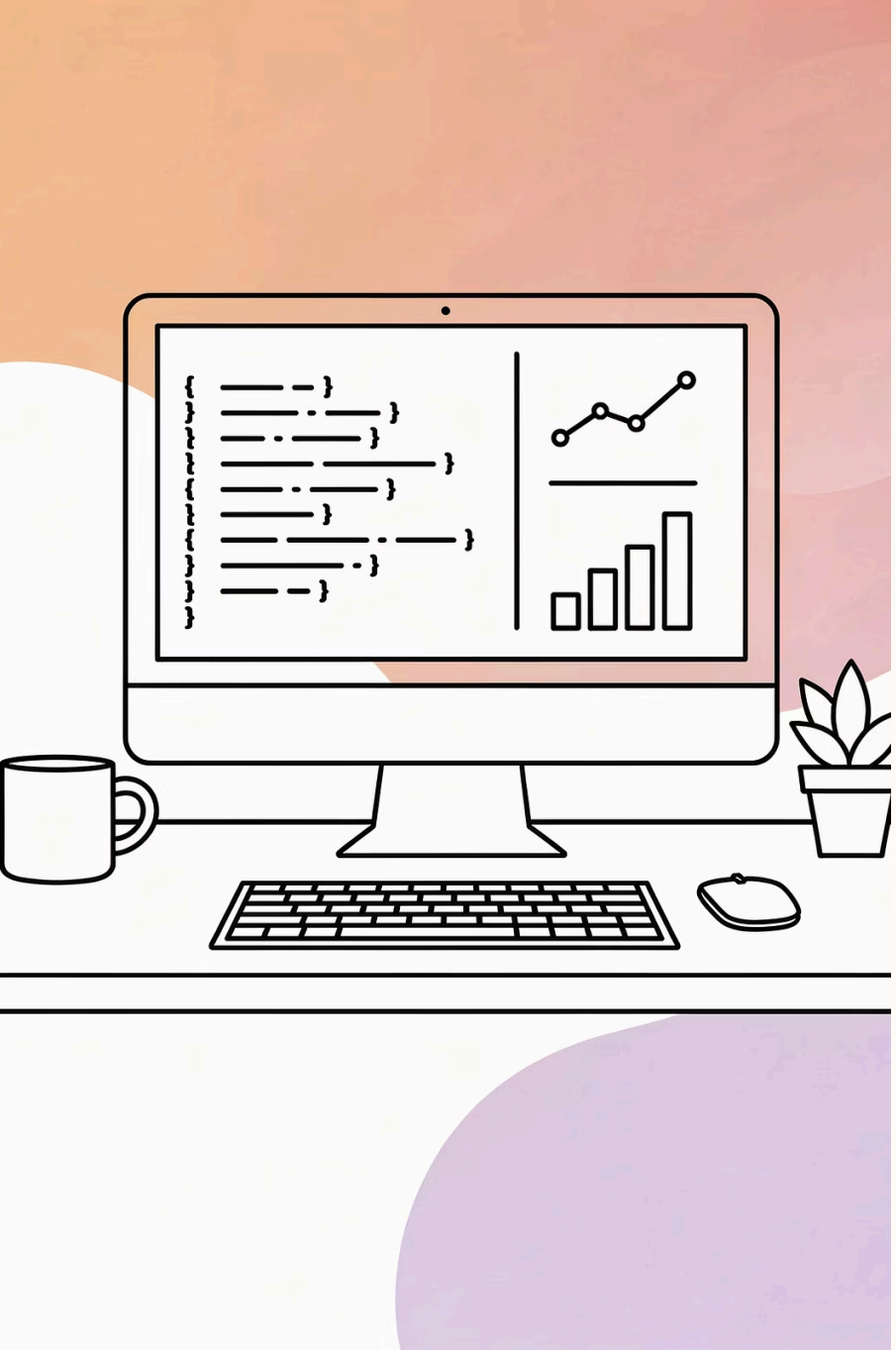


CEFET/RJ

Ordenação em Python

Ordenar dados significa reorganizar valores para facilitar comparação, análise e decisões. Nesta aula vamos trabalhar com vetores e tabelas usando NumPy e pandas.

Eduardo Ogasawara

eduardo.ogasawara@cefet-rj.br

https://eic.cefet-rj.br/~eogasawara

PREPARAÇÃO

Criando Vetores de Dados

Vamos criar três vetores: peso, altura e identificador do sujeito. Em Python, dados numéricos são representados por arrays do NumPy.

```
weight = np.array([60, 72, 57, 90, 95, 72])
height = np.array([1.75, 1.80, 1.65, 1.90, 1.74, 1.91])
subject = ["A", "B", "C", "D", "E", "F"]
```

weight

Vetor de pesos em kg

height

Vetor de alturas em metros

subject

Identificadores dos sujeitos

DATAFRAME

Construindo um DataFrame

Quando dados estão relacionados, usamos um DataFrame. Cada linha representa um sujeito e cada coluna uma variável.

```
d = pd.DataFrame({
    "weight": weight,
    "height": height,
    "subject": subject
})
```

weight	height	subject
60	1.75	A
72	1.80	B
57	1.65	C
90	1.90	D
95	1.74	E

Ordenar Valores vs Obter a Ordem

np.sort

Devolve os valores ordenados

```
array([1.65, 1.74, 1.75,
       1.80, 1.90, 1.91])
```

np.argsort

Devolve os índices da ordenação

```
array([2, 4, 0, 1, 3, 5])
```

np.sort reorganiza apenas um vetor. np.argsort mostra as posições que produzem essa ordenação.

CONCEITO

Interpretando argsort

01

Posição Original

Cada número indica a posição original do próximo menor valor

02

Reorganização

Explica como o vetor foi reorganizado

03

Reprodução

Esses índices são úteis para entender ou reproduzir a ordem

```
idx = np.argsort(height)
print(idx)
# Saída: [2 4 0 1 3 5]
```

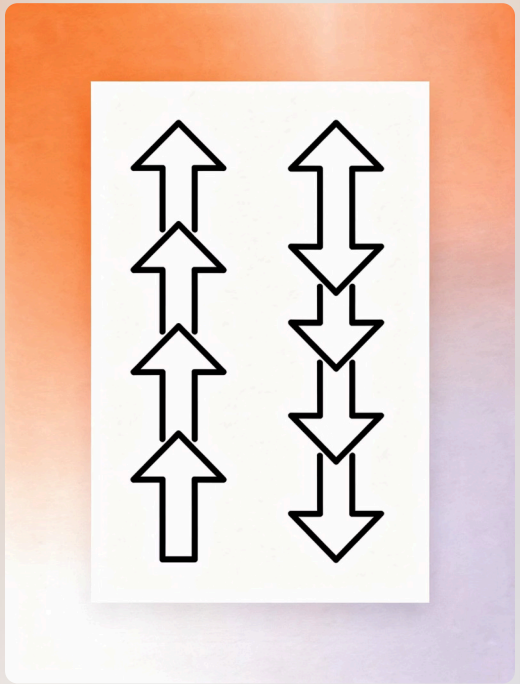
PANDAS

Ordenando uma Tabela Completa

Em pandas usamos `sort_values` para ordenar tabelas. As linhas são reorganizadas com base em uma coluna e todas as colunas permanecem alinhadas.

```
ds = d.sort_values("height")
print(ds)
```

weight	height	subject
57	1.65	C
95	1.74	E
60	1.75	A
72	1.80	B
90	1.90	D
72	1.91	F



Ordenação Crescente e Decrescente

Crescente (padrão)

```
d.sort_values("height",
              ascending=True)
```

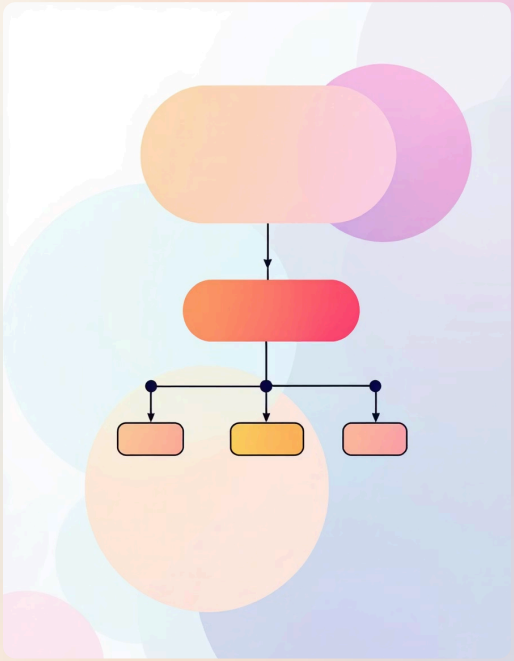
Do menor para o maior valor

Decrescente

```
d.sort_values("height",
              ascending=False)
```

Do maior para o menor valor

Ordenação decrescente é comum em rankings e comparações. Basta usar o parâmetro ascending.



AVANÇADO

Ordenação por Múltiplas Colunas

1 **Critério Primário**

Primeiro ordena por altura

2 **Critério Secundário**

Depois por peso como desempate

3 **Ordenação Hierárquica**

Cria uma ordenação em níveis

```
d.sort_values(["height", "weight"])
```

Uma única coluna pode não ser suficiente. Podemos usar outra coluna como critério de desempate.

Conceitos-Chave

1

np.sort

Vetores são ordenados com np.sort

```
np.sort(height)
```

2

np.argsort

Índices de ordenação são obtidos com np.argsort

```
np.argsort(height)
```

3

sort_values

Tabelas são ordenadas com DataFrame.sort_values

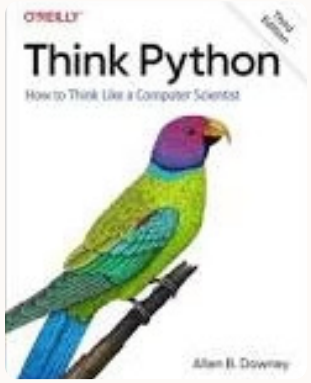
```
d.sort_values("height")
```

4

Alinhamento

Todas as colunas permanecem alinhadas

Referências



Think Python

Downey, A. *Think Python: How to Think Like a Computer Scientist.* O'Reilly Media.

An essential introduction to programming fundamentals and computational thinking using Python.

Python Data Science Handbook

VanderPlas, J. *Python Data Science Handbook.* O'Reilly Media.

A comprehensive guide to essential tools for working with data in Python, including NumPy, Pandas, and visualization libraries.

Data Science from Scratch

Grus, J. *Data Science from Scratch.* O'Reilly Media.

Learn data science fundamentals by building algorithms and tools from the ground up using Python.