Educação Profissional Paulista

Técnico em **Ciência de Dados**

Secretaria da Educação

SÃO PAULO
GOVERNO DO ESTADO

Bibliotecas: Pandas, NumPy, SciPy, Matplotlib e Seaborn

Pandas: combinando *DataFrames*

Aula 2

Código da aula: [DADOS]ANO1C2B4S26A2

Mapa da Unidade 5 Componente 3

Bibliotecas: Pandas, NumPy, SciPy, Matplotlib e Seaborn

semana 23

Pandas: acesso e seleção

semana 24

Pandas: ler e escrever

semana 25

Pandas: transformando DataFrame

Você está aqui!

semana 26

Pandas: combinando DataFrames

Bibliotecas: Pandas, NumPy, SciPy, Matplotlib e Seaborn

Mapa da Unidade 5 Componente 3

Você está aqui!

26

Pandas: combinando DataFrames

Aula 2

Código da aula: [DADOS]ANO1C2B4S26A2

Objetivos da Aula

- Praticar seleção de linhas e colunas no DataFrame da biblioteca Pandas do Python.

Recursos Didáticos

- Recurso audiovisual para exibição de vídeos e imagens;
- Acesso ao laboratório de informática e/ou internet;
- Software Anaconda/Jupyter Notebook instalado ou similar.

Duração da Aula

50 minutos.

Competências Técnicas

- Ser proficiente em linguagens de programação para manipular e analisar grandes conjuntos de dados;
- Usar técnicas para explorar e analisar dados, aplicar modelos estatísticos, identificar padrões, realizar inferências e tomar decisões baseadas em evidências.

Competências Socioemocionais

- Colaborar efetivamente com outros profissionais, como cientistas de dados e engenheiros de dados;
- Trabalhar em equipes multifuncionais, colaborando com colegas, gestores e clientes.

Construindo o **conceito**

Exemplos: filtrando linhas e colunas

Com o DataFrame abaixo, selecione as pessoas que têm idade de 30 anos.

```
import pandas as pd

# Criando um DataFrame de clientes
dados = {
    "nome": ["João", "Maria", "Pedro", "Ana"],
    "idade": [30, 25, 22, 32],
    "cidade": ["São Paulo", "Rio de Janeiro", "Belo Horizonte", "Salvador"],
}
df = pd.DataFrame(dados)
df
```

	nome	idade	cidade
0	João	30	São Paulo
1	Maria	25	Rio de Janeiro
2	Pedro	22	Belo Horizonte
3	Ana	32	Salvador

```
# Selecionando linhas com idade 30
df_filtrado = df[df["idade"] == 30]

# Mostrando o DataFrame filtrado
df_filtrado
```

	nome	idade	cidade
0	João	30	São Paulo

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Construindo o **conceito**

Exemplos: filtrando linhas e colunas

Com o DataFrame abaixo, selecione os produtos que têm preço entre 50 e 100 e data maior que 2023.

```
import pandas as pd

# Criando um DataFrame de vendas
dados = {
    "produto": ["Notebook", "Celular", "Camisa", "Tenis"],
    "preço": [2500, 800, 50, 120],
    "data": ["2023-01-10", "2023-03-15", "2023-05-22", "2023-04-01"],
}
df = pd.DataFrame(dados)
df
```

	produto	preço	data
0	Notebook	2500	2023-01-10
1	Celular	800	2023-03-15
2	Camisa	50	2023-05-22
3	Tenis	120	2023-04-01

```
# Selecionando linhas com preço entre 50 e 100 e data em 2023
df_filtrado = df[(df["preço"] >= 50) & (df["preço"] <= 100) & (df["data"] >= "2023-01-01")]

# Mostrando o DataFrame filtrado
df_filtrado
```

	produto	preço	data
2	Camisa	50	2023-05-22

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Construindo o **conceito**

Exemplos: filtrando linhas e colunas

Com o DataFrame abaixo, selecione as linhas com data entre **01/01/2024** e **31/03/2024** e que tenham **valor superior a 1.000**.

```python
import pandas as pd

# Criando um DataFrame de transações bancárias
dados = {
    "data": ["2024-02-01", "2024-03-12", "2024-04-20", "2024-01-30"],
    "valor": [1500, 850, 220, 5000],
    "tipo": ["Débito", "Crédito", "Débito", "Crédito"],
}
df = pd.DataFrame(dados)
df
```

	data	valor	tipo
0	2024-02-01	1500	Débito
1	2024-03-12	850	Crédito
2	2024-04-20	220	Débito
3	2024-01-30	5000	Crédito

```python
# Definindo as datas inicial e final
data_inicial = "2024-01-01"
data_final = "2024-03-31"

# Selecionando linhas com data entre data_inicial e data_final e valor superior a 1000
df_filtrado = df[
    (df["data"] >= data_inicial) & (df["data"] <= data_final) & (df["valor"] > 1000)
]

# Mostrando o DataFrame filtrado
df_filtrado
```

	data	valor	tipo
0	2024-02-01	1500	Débito
3	2024-01-30	5000	Crédito

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Construindo
o **conceito**

Exemplos: filtrando linhas e colunas

Com o DataFrame abaixo, selecione as linhas com cargo **“Gerente” ou “Diretor”.**

```python
import pandas as pd

# Criando um DataFrame de funcionários
dados = {
    "nome": ["Carlos", "Ana", "Marcos", "Fernanda"],
    "cargo": ["Gerente", "Atendente", "Vendedor", "Diretora"],
    "salário": [5000, 2500, 3200, 7000],
}
df = pd.DataFrame(dados)
df
```

	nome	cargo	salário
0	Carlos	Gerente	5000
1	Ana	Atendente	2500
2	Marcos	Vendedor	3200
3	Fernanda	Diretora	7000

```python
# Selecionando linhas com cargo "Gerente" ou "Diretor"
cargos_desejados = ["Gerente", "Diretor"]
df_filtrado = df[df["cargo"].isin(cargos_desejados)]

# Mostrando o DataFrame filtrado
df_filtrado
```

	nome	cargo	salário
0	Carlos	Gerente	5000

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Construindo
o **conceito**

Exemplos: filtrando linhas e colunas

Com o DataFrame abaixo, selecione as linhas com nota1 maior que 7 e nota2 maior que 8.

```python
import pandas as pd

# Criando um DataFrame de alunos
dados = {
    "nome": ["João", "Maria", "Pedro", "Ana"],
    "curso": ["Engenharia", "Medicina", "Direito", "Ciências da Computação"],
    "nota1": [8.5, 9.2, 7.8, 9.1],
    "nota2": [9.0, 8.8, 8.2, 9.4],
}
df = pd.DataFrame(dados)
df
```

	nome	curso	nota1	nota2
0	João	Engenharia	8.5	9.0
1	Maria	Medicina	9.2	8.8
2	Pedro	Direito	7.8	8.2
3	Ana	Ciências da Computação	9.1	9.4

```python
# Selecionando linhas com nota1 > 7 e nota2 > 8
df_filtrado = df[(df["nota1"] > 7) & (df["nota2"] > 8)]

# Mostrando o DataFrame filtrado
df_filtrado
```

	nome	curso	nota1	nota2
0	João	Engenharia	8.5	9.0
1	Maria	Medicina	9.2	8.8
2	Pedro	Direito	7.8	8.2
3	Ana	Ciências da Computação	9.1	9.4

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Construindo o **conceito**

Exemplos: filtrando linhas e colunas

Com o DataFrame abaixo, selecione as linhas com preço menor que 50 ou estoque zero.

```python
import pandas as pd

# Criando um DataFrame de produtos
dados = {
    "nome": ["Camisa", "Calça", "Tênis", "Boné"],
    "categoria": ["Vestuário", "Vestuário", "Calçados", "Acessórios"],
    "preço": [35, 80, 120, 25],
    "estoque": [10, 5, 2, 0],
}
df = pd.DataFrame(dados)
df
```

	nome	categoria	preço	estoque
0	Camisa	Vestuário	35	10
1	Calça	Vestuário	80	5
2	Tênis	Calçados	120	2
3	Boné	Acessórios	25	0

```python
# Selecionando linhas com preço menor que 50 ou estoque zero
df_filtrado = df[(df["preço"] < 50) | (df["estoque"] == 0)]

# Mostrando o DataFrame filtrado
df_filtrado
```

	nome	categoria	preço	estoque
0	Camisa	Vestuário	35	10
3	Boné	Acessórios	25	0

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Construindo o **conceito**

Exemplos: filtrando linhas e colunas

Com o DataFrame abaixo, selecione as linhas com população menor ou igual a 100.000.

```python
import pandas as pd

# Criando um DataFrame de cidades
dados = {
    "nome": ["São Paulo", "Rio de Janeiro", "Belo Horizonte", "Salvador"],
    "estado": ["SP", "RJ", "MG", "BA"],
    "população": [12.2, 6.7, 5.5, 2.9],
}
df = pd.DataFrame(dados)
df
```

	nome	estado	população
0	São Paulo	SP	12.2
1	Rio de Janeiro	RJ	6.7
2	Belo Horizonte	MG	5.5
3	Salvador	BA	2.9

```python
# Selecionando linhas com população menor ou igual a 100.000
df_filtrado = df[df["população"] <= 100000]

# Mostrando o DataFrame filtrado
df_filtrado
```

	nome	estado	população
0	São Paulo	SP	12.2
1	Rio de Janeiro	RJ	6.7
2	Belo Horizonte	MG	5.5
3	Salvador	BA	2.9

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Construindo o **conceito**

Exemplos: filtrando linhas e colunas

Com o DataFrame abaixo, selecione as colunas “nome”, “idade” e “cidade”.

```python
import pandas as pd

# Criando um DataFrame de exemplo
dados = {
    "nome": ["João", "Maria", "Pedro", "Ana"],
    "idade": [30, 25, 22, 32],
    "cidade": ["São Paulo", "Rio de Janeiro", "Belo Horizonte", "Salvador"],
    "curso": ["Engenharia", "Medicina", "Direito", "Ciências da Computação"],
}
df = pd.DataFrame(dados)
df
```

	nome	idade	cidade	curso
0	João	30	São Paulo	Engenharia
1	Maria	25	Rio de Janeiro	Medicina
2	Pedro	22	Belo Horizonte	Direito
3	Ana	32	Salvador	Ciências da Computação

```python
# Selecionando colunas específicas
colunas_desejadas = ["nome", "idade", "cidade"]
df_selecionado = df.loc[:, colunas_desejadas]

# Mostrando o DataFrame com colunas selecionadas
df_selecionado
```

	nome	idade	cidade
0	João	30	São Paulo
1	Maria	25	Rio de Janeiro
2	Pedro	22	Belo Horizonte
3	Ana	32	Salvador

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Construindo o **conceito**

Exemplos: filtrando linhas e colunas

Com o DataFrame abaixo, selecione as linhas com idade maior que 25 e colunas “nome”, “cidade” e “nota2”.

```python
import pandas as pd

# Criando um DataFrame de exemplo
dados = {
    "nome": ["João", "Maria", "Pedro", "Ana"],
    "idade": [30, 25, 22, 32],
    "cidade": ["São Paulo", "Rio de Janeiro", "Belo Horizonte", "Salvador"],
    "curso": ["Engenharia", "Medicina", "Direito", "Ciências da Computação"],
    "nota1": [8.5, 9.2, 7.8, 9.1],
    "nota2": [9.0, 8.8, 8.2, 9.4],
}
df = pd.DataFrame(dados)
df
```

	nome	idade	cidade	curso	nota1	nota2
0	João	30	São Paulo	Engenharia	8.5	9.0
1	Maria	25	Rio de Janeiro	Medicina	9.2	8.8
2	Pedro	22	Belo Horizonte	Direito	7.8	8.2
3	Ana	32	Salvador	Ciências da Computação	9.1	9.4

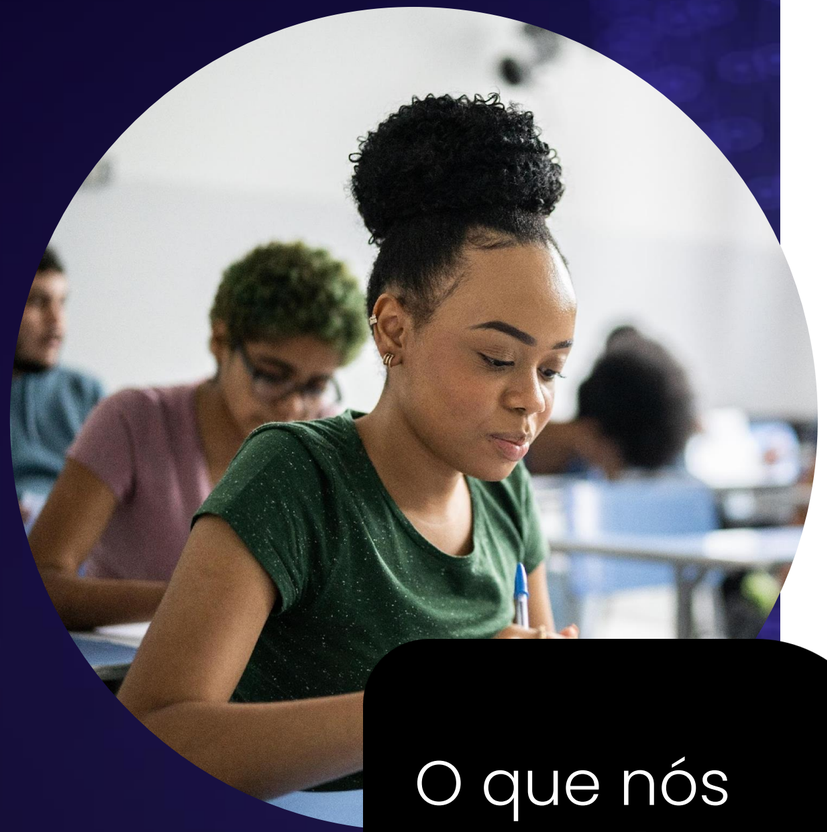
```python
# Selecionando linhas com idade maior que 25 e colunas 'nome', 'cidade' e 'nota2'
df_selecionado = df.loc[(df["idade"] > 25), ["nome", "cidade", "nota2"]]

# Mostrando o DataFrame com linhas e colunas selecionadas
df_selecionado
```

	nome	cidade	nota2
0	João	São Paulo	9.0
3	Ana	Salvador	9.4

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

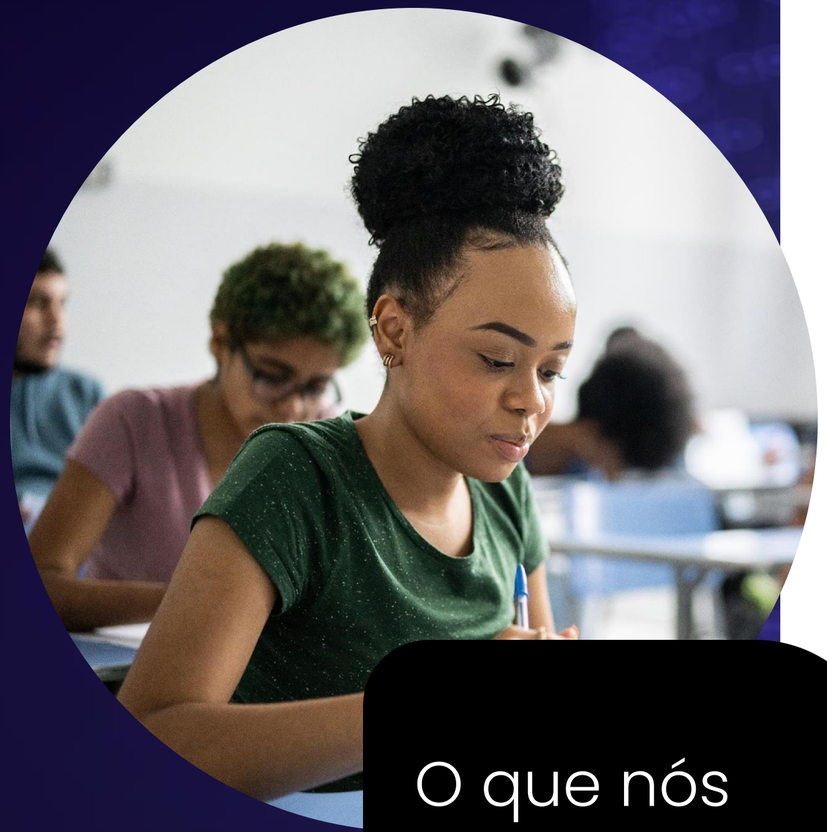
Diálogo



© Getty Images

O que nós **aprendemos hoje?**

Nuvem de palavras



© Getty Images

O que nós **aprendemos hoje?**

Então ficamos assim...

1. Praticamos a seleção/filtro de linhas usando a biblioteca Pandas do Python.

2. Vimos formas diferentes de realizar a mesma operação de seleção e filtro de informações do DataFrame.

3. Praticamos como trabalhar com filtro de colunas usando Pandas.

Saiba mais

Que tal encarar o desafio de aprender Pandas em 10 minutos?

Acesse o guia abaixo e traduza para o português para você saber tudo de Pandas em 10 minutos!

PANDAS. *Viewing data*, [s.d.]. Disponível em: https://pandas.pydata.org/pandas-docs/stable/user_guide/10min.html#viewing-data. Acesso em: 19 jul. 2024.

Referências da aula

MCKINNEY, W. *Python para análise de dados*: tratamento de dados com Pandas, NumPy & Jupyter. São Paulo: Novatec, 2023.

PANDAS. *Pandas documentation*, 10 abr. 2024. Disponível em: https://pandas.pydata.org/docs/. Acesso em: 19 jul. 2024.

Educação Profissional Paulista

Técnico em **Ciência de Dados**

Secretaria da Educação

SÃO PAULO
GOVERNO DO ESTADO