

1. Criando DataFrames de exemplo

Antes de começar a combinar tabelas, crie manualmente dois DataFrames de exemplo.

python

```
import pandas as pd

df1 = pd.DataFrame({
    'Nome': ['João', 'João', 'Pedro', 'Caio'],
    'Telefone': [12121, 343434, 565656, 787878],
    'Carros': ['azul', 'preto', 'verde', 'amarelo']
})

df2 = pd.DataFrame({
    'Nome': ['João', 'Marcelo', 'Thiago', 'Caio'],
    'Irmãos': [1, 3, 2, 2]
})
```

Esses DataFrames serão usados para testar diferentes maneiras de combinar dados.

2. Combinando DataFrames com merge

O método `merge()` do Pandas funciona como um “cruzamento” de tabelas, semelhante ao SQL. Ele usa uma coluna chave para relacionar os dados.

Exemplo básico:

python

```
df_combinado = pd.merge(df1, df2, how='inner', on='Nome')
print(df_combinado)
```

Saída esperada:

text

	Nome	Telefone	Carros	Irmãos
0	João	12121	azul	1
1	João	343434	preto	1
2	Caio	787878	amarela	2

Tipos de junção possíveis:

Tipo de junção	Descrição
inner	Mostra apenas valores comuns nas duas tabelas
left	Mantém todas as linhas do DataFrame da esquerda
right	Mantém todas as linhas do DataFrame da direita
outer	Junta todas as linhas das duas tabelas, preenchendo com NaN onde faltar dado

3. Concatenando DataFrames

O método `concat()` serve para juntar tabelas, mas sem precisar de colunas em comum.

```
python
```

```
df_concat = pd.concat([df1, df2], axis=0, ignore_index=True)
print(df_concat)
```

O parâmetro `axis=0` junta as tabelas na vertical (uma embaixo da outra).
Se quiser juntar lado a lado, use `axis=1`.

Exemplo:

```
python
```

```
df_concat_horizontal = pd.concat([df1, df2], axis=1)
print(df_concat_horizontal)
```

4. Exemplo prático — Clientes e pedidos

Imaginando uma loja com dois conjuntos de dados: clientes e pedidos, podemos combinar as informações pelo ID do cliente.

python

```
clientes = pd.DataFrame({
    'IDcliente': [1, 2, 3],
    'nome': ['João', 'Maria', 'Pedro'],
    'idade': [30, 25, 22]
})

pedidos = pd.DataFrame({
    'IDcliente': [1, 1, 2, 3],
    'produto': ['Camisa', 'Tênis', 'Celular', 'Notebook'],
    'valor': [50, 120, 800, 2500]
})
```

Combine as tabelas:

python

```
df_final = pd.merge(clientes, pedidos, on='IDcliente', how='inner')
print(df_final)
```

Resultado:

text

	IDcliente	nome	idade	produto	valor
0	1	João	30	Camisa	50
1	1	João	30	Tênis	120
2	2	Maria	25	Celular	800
3	3	Pedro	22	Notebook	2500

5. Merge completo (outer join)

python

```
df_merge = pd.merge(clientes, pedidos, on='IDcliente', how='outer')
print(df_merge)
```

Esse tipo de junção inclui todos os registros, mesmo se não houver correspondência, preenchendo com **NaN** os campos vazios.

6. Exemplo final — Sorveteria Soops Smiles

Agora, vamos aplicar tudo em um exemplo completo.

Imagine que você é gerente de uma sorveteria e precisa unir duas tabelas de vendas e depois somar receitas por sabor.

Passo 1 – Criar as duas tabelas de vendas

python

```
vendas = pd.DataFrame({
    'Data': pd.date_range(start='2024-05-01', periods=10, freq='D'),
    'QuantidadeVendida': [48, 38, 24, 17, 30, 48, 28, 32, 20, 20],
    'Sabor': ['Baunilha', 'Creme', 'Creme', 'Creme', 'Baunilha', 'Baunilha', 'Baunilha',
})

vendas_novas = pd.DataFrame({
    'Data': pd.date_range(start='2024-05-11', periods=10, freq='D'),
    'QuantidadeVendida': [33, 45, 49, 33, 12, 31, 11, 33, 39, 47],
    'Sabor': ['Baunilha', 'Morango', 'Baunilha', 'Creme', 'Baunilha', 'Creme', 'Creme',
})
```

Passo 2 – Unir as duas tabelas

python

```
vendas_completas = pd.concat([vendas, vendas_novas], ignore_index=True)
```

Passo 3 – Adicionar tabela de preços

python

```
precos = pd.DataFrame({
    'Sabor': ['Baunilha', 'Chocolate', 'Morango', 'Creme'],
    'Preco': [5.0, 6.0, 5.5, 4.5]
})

vendas_completas = pd.merge(vendas_completas, precos, on='Sabor', how='left')
```

Passo 4 – Calcular receita e filtrar resultados

python

```
vendas_completas['Receita'] = vendas_completas['QuantidadeVendida'] * vendas_completas['PrecoUnitario']
filtro = vendas_completas[vendas_completas['Receita'] > 200]
print(filtro)
```

O resultado mostrará apenas as linhas onde a receita foi superior a R\$ 200.

7. Desafio

- . Filtre apenas as vendas de Baunilha.
- . Calcule a receita média por sabor.
- . Ordene as vendas por data.

Conclusão

Neste tutorial, você aprendeu a:

Combinar tabelas com `merge` e `concat`.

Escolher o tipo de junção ideal (`inner`, `left`, `right`, `outer`).

Aplicar filtros e criar colunas calculadas em DataFrames.

Essas técnicas são fundamentais para construir análises completas e preparar dados para visualizações ou relatórios.