O que significa a expressão df.groupby(['Região','Produto'])['Vendas'].sum().reset\_index()

Vamos destrinchar passo a passo a expressão:

df.groupby(['Região','Produto'])['Vendas'].sum().reset\_index()

Aqui estamos lidando com **Pandas**, uma biblioteca muito usada em Python para manipulação de dados.

**1. df.groupby(['Região','Produto'])**

* df é o **DataFrame**, ou seja, a tabela de dados.
* groupby(['Região','Produto']) significa **agrupar os dados** pelas colunas Região e Produto.
* Ou seja, todas as linhas que tiverem a mesma combinação de Região e Produto vão ser consideradas um **grupo**.

Exemplo:

| **Região** | **Produto** | **Vendas** |
| --- | --- | --- |
| Norte | A | 100 |
| Norte | A | 150 |
| Sul | B | 200 |
| Norte | B | 50 |

Depois do groupby(['Região','Produto']), temos grupos como:

* Grupo 1: Região = Norte, Produto = A → [100, 150]
* Grupo 2: Região = Sul, Produto = B → [200]
* Grupo 3: Região = Norte, Produto = B → [50]

**2. ['Vendas'].sum()**

* Aqui estamos selecionando a coluna Vendas dentro de cada grupo.
* sum() vai **somar os valores de vendas de cada grupo**.

Continuando o exemplo:

* Grupo 1: 100 + 150 = 250
* Grupo 2: 200
* Grupo 3: 50

O resultado agora é algo como:

| **Região** | **Produto** | **Vendas** |
| --- | --- | --- |
| Norte | A | 250 |
| Sul | B | 200 |
| Norte | B | 50 |

**3. .reset\_index()**

* Depois de usar groupby, as colunas usadas para agrupar (Região e Produto) viram **índice** do DataFrame.
* reset\_index() transforma esses índices de volta em **colunas normais**, deixando o DataFrame “bonitinho” de novo.

✅ **Resumo completo:**  
A expressão:

df.groupby(['Região','Produto'])['Vendas'].sum().reset\_index()

Faz o seguinte:

1. Agrupa os dados por Região e Produto.
2. Soma as vendas de cada grupo.
3. Retorna um DataFrame com Região, Produto e a soma das Vendas como colunas normais.

.reset\_index() no contexto do **pandas**.

**Contexto**

Quando você faz um groupby em um DataFrame, como por exemplo:

import pandas as pd

df = pd.DataFrame({

'Região': ['Norte', 'Norte', 'Sul', 'Sul'],

'Produto': ['A', 'B', 'A', 'B'],

'Vendas': [100, 200, 150, 250]

})

df\_grouped = df.groupby(['Região', 'Produto']).sum()

print(df\_grouped)

O resultado será:

Vendas

Região Produto

Norte A 100

B 200

Sul A 150

B 250

**Perceba que Região e Produto agora são índices do DataFrame** (não são mais colunas normais).

**O que .reset\_index() faz**

Se quisermos voltar a ter **Região** e **Produto** como colunas “normais”, usamos:

df\_reset = df\_grouped.reset\_index()

print(df\_reset)

O resultado será:

Região Produto Vendas

0 Norte A 100

1 Norte B 200

2 Sul A 150

3 Sul B 250

Agora o DataFrame está “bonitinho” de novo, pronto para manipulação ou exportação, com índices padrão (0, 1, 2, 3…) e todas as colunas visíveis.

**Resumo**

* **groupby()** → coloca as colunas usadas para agrupar como **índices**.
* **reset\_index()** → transforma esses índices de volta em **colunas normais**, deixando o DataFrame organizado.