

08 **Criando subplots em duas direções #2**

PRÓXIMA ATIVIDADE

**Transcrição**

Agora nós temos todas as informações no gráfico, é possível entender o que temos no eixo x, no eixo y, qual é cada um dos países, e já podemos começar a refletir sobre algumas coisas.

Criando subplots em duas direções

Perguntamos anteriormente se as tendências de imigração eram as mesmas para todos os países. Analisando os gráficos, conseguimos entender que não.

Por exemplo, no Brasil, temos dois picos maiores: o primeiro entre 1990 e 1995, e depois um grande pico entre 2005 e 2010. Já na Colômbia, não há picos no começo, mas há um grande aumento a partir do final dos anos 90. Então, depois de 1995, temos um aumento bem grande, que aconteceu antes do Brasil.

Outro ponto de atenção é que o número de imigrantes máximo no Brasil está em 2.500, enquanto na Colômbia está em 6.000 imigrantes. Temos uma expressividade bem maior na Colômbia.

Já na Argentina, temos dois picos maiores: um entre 1990 e 1995, como no Brasil, e o segundo pico acontecendo também um pouco antes do Brasil, começando nos anos 2000. No Peru, temos dois picos, um começando em 1985 e terminando no final dos anos 90, e outro começando também nos anos 2000, com um maior valor em cerca de 2010.

Precisamos fazer um destaque importante: esses gráficos estão em escalas diferentes de eixo y. No gráfico do Brasil, o valor vai até 2500; nos da Argentina e do Peru, até 1500; e no da Colômbia, até 6000. Isso pode gerar uma confusão para quem visualiza o gráfico.

Se o gráfico for publicado ou apresentado em algum meio, podemos passar a impressão de que ele está **distorcido**, pois visualmente, podemos interpretar que os números do Brasil e da Colômbia são praticamente os mesmos no valor máximo. Porém, os valores são diferentes nos eixos.

Ajustando o eixo y dos gráficos

Podemos consertar esses valores do eixo y, deixando todos os gráficos com o mesmo valor mínimo e o mesmo valor máximo. Para fazer isso, criaremos mais uma instrução `for`.

Antes da função `plt.show()`, adicionaremos o valor mínimo do eixo y, então vamos definir o parâmetro `ymin` sendo igual a 0. Em seguida, vamos adicionar o valor máximo, então `ymax`. Nesse caso, o valor

pode ser em torno de 7000, por exemplo.

```
# Código suprimido
```

```
ymin = 0
```

```
ymax = 7000
```

```
# Código suprimido
```



COPIAR CÓDIGO

Agora vamos incluir a instrução `for` para ajustar os valores. Após a estrutura `for ax in axs`, vamos adicionar o método `ravel()`, que consegue transformar um *array* bidimensional em unidimensional e fazer algumas aplicações. Finalizaremos o método com dois-pontos. Com isso, vamos conseguir iterar sobre todos os objetos.

Na linha de código abaixo, vamos adicionar a função `ax_set_ylim()`, referente ao **limite**. Entre os parênteses da função, passaremos as variáveis `ymin` e `ymax`, que serão os limites do eixo y.

```
ymin = 0
```

```
ymax = 7000
```

```
for ax in axs.ravel():  
    ax.set_ylim(ymin, ymax)
```

[COPIAR CÓDIGO](#)

Chegaremos ao seguinte código:

```
fig, axs = plt.subplots(2,2, figsize=(10,6))  
fig.subplots_adjust(hspace=0.5, wspace=0.3)  
  
axs[0,0].plot(df.loc['Brasil', anos])  
axs[0,0].set_title('Brasil')  
  
axs[0,1].plot(df.loc['Colômbia', anos])  
axs[0,1].set_title('Colômbia')  
  
axs[1,0].plot(df.loc['Argentina', anos])  
axs[1,0].set_title('Argentina')  
  
axs[1,1].plot(df.loc['Peru', anos])  
axs[1,1].set_title('Peru')  
  
for ax in axs.flat:
```

```
ax.xaxis.set_major_locator(plt.MultipleLocator(5))

for ax in axs.flat:
    ax.set_xlabel('Ano')
    ax.set_ylabel('Número de imigrantes')

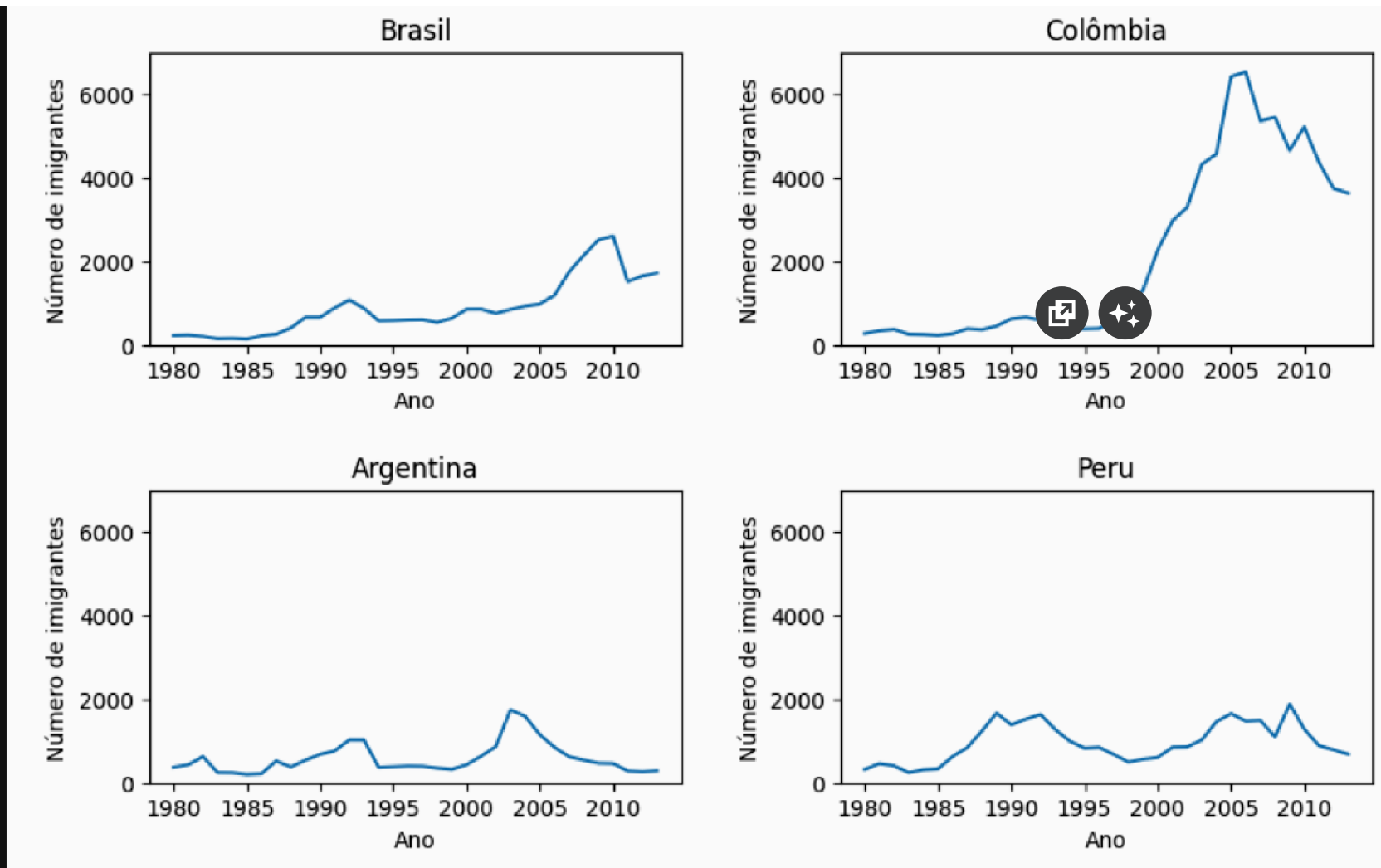
ymin = 0
ymax = 7000

for ax in axs.ravel():
    ax.set_ylim(ymin, ymax)

plt.show()
```

[COPIAR CÓDIGO](#)

Ao executar a célula, teremos todos os eixos y com os mesmos valores mínimo e máximo:



Agora conseguimos entender que a Colômbia tem valores muito mais altos em relação aos demais países e nossos gráficos estão praticamente prontos.

Adicionando um título geral

A última coisa que podemos adicionar é um **título geral** para entender do que se trata a figura completa. Para isso, existe uma função específica chamada `suptitle()` que adicionaremos na terceira linha de código.

Começaremos digitando a variável `fig` seguida da função `suptitle()`, para a qual passaremos o título geral entre aspas simples. O título será "Imigração dos quatro maiores países da América do Sul para o Canadá de 1980 a 2013".

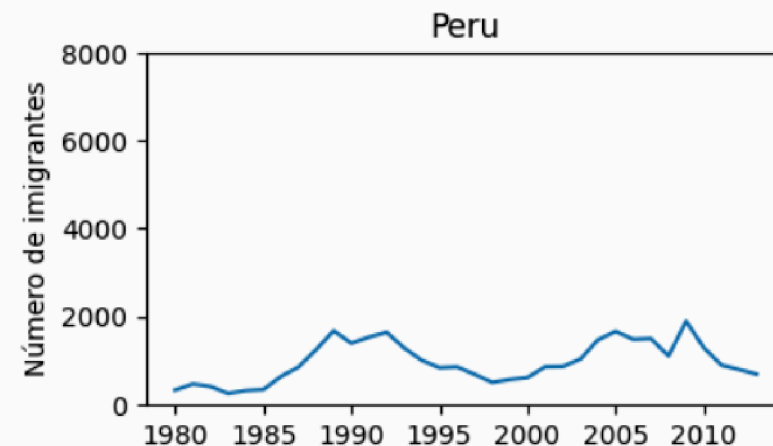
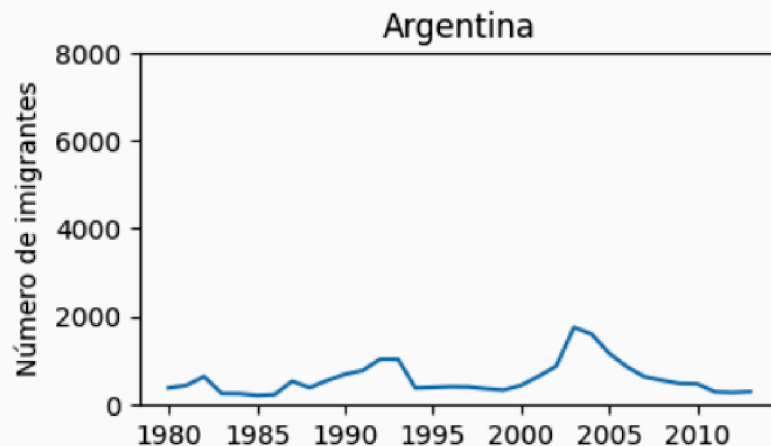
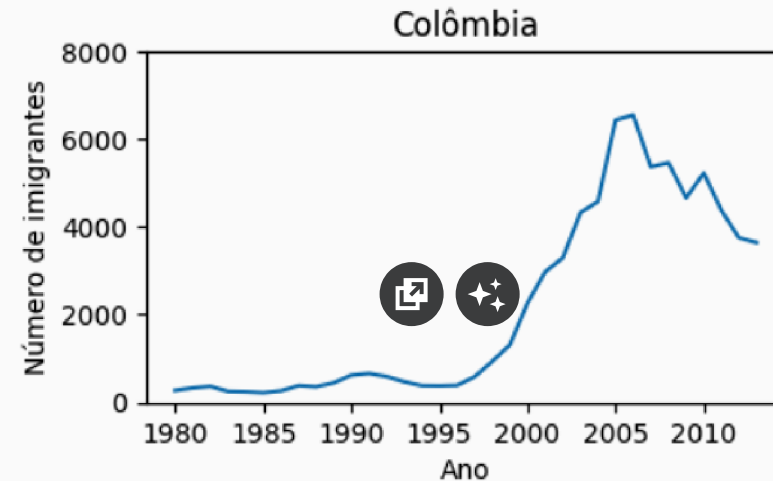
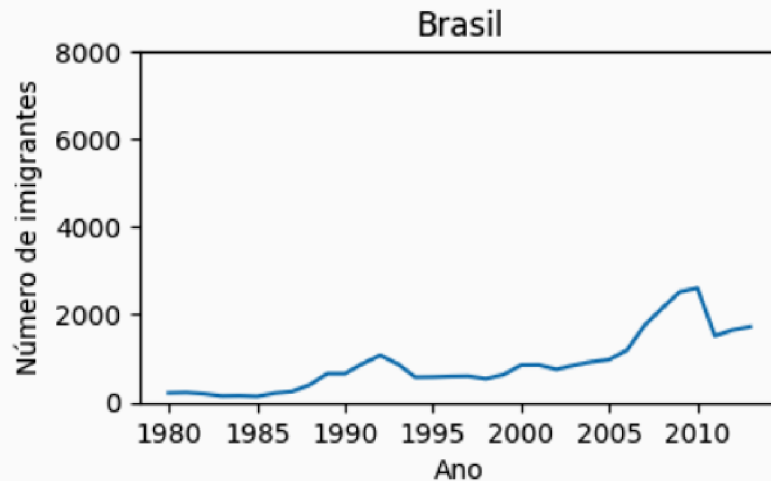
```
fig, axs = plt.subplots(2,2, figsize=(10,6))  
fig.subplots_adjust(hspace=0.5, wspace=0.3)  
fig.suptitle('Imigração dos quatro maiores países da América do Sul para o Canadá d
```

Código suprimido

COPIAR CÓDIGO

Uma vez executada a célula, teremos nossa figura final contendo um título geral e 4 subplots com as informações de cada país representado.

Imigração dos quatro maiores países da América do Sul para o Canadá de 1980 a 2013



Conclusão

Além de criar subplots e construir figuras, a biblioteca **Matplotlib** oferece uma ampla gama de opções de personalização para os gráficos. Daqui em diante, começaremos a personalizá-los alterando cores e

fazendo uma série de outras coisas bem legais!

