



07 Criando subplots em duas direções #1



PRÓXIMA ATIVIDADE



Transcrição

Agora vamos seguir para outra etapa do nosso projeto: analisar as tendências de imigração para os países da América do Sul, incluindo o Brasil. Será possível analisar mais de um país simultaneamente?

Criando subplots em duas direções

Podemos começar analisando os 4 maiores países da América do Sul: Brasil, Colômbia, Argentina e Peru. Nosso objetivo é comparar as tendências **em uma única figura**, utilizando subplots.

No vídeo anterior, fizemos uma figura com 2 subplots, mas agora teremos uma figura com 4. Como podemos fazer isso? Será que as tendências de imigração foram diferentes para cada um dos países? Descobriremos isso em breve!

Criando a figura

Para **criar a figura**, vamos digitar a variável `fig` em uma nova célula, seguida de `axs`, separadas por uma vírgula. Essas variáveis serão iguais à função `plt.subplots()`. Entre parênteses, passaremos quantos subplots teremos na figura. Nesse caso, queremos 2 subplots em uma primeira linha, e outros 2 subplots em uma segunda linha. Então, digitamos `2,2`.

Além disso, vamos definir o **tamanho da figura**. Para isso, usamos o parâmetro `figsize`, que será igual aos valores `10,6`, indicados entre parênteses após o sinal de `=`.

```
fig, axs = plt.subplots(2,2, figsize=(10,6))
```

[COPIAR CÓDIGO](#)

Criando o subplot do Brasil

Na próxima linha de código, começaremos a criar os subplots para cada um dos países. Começaremos pelo **Brasil**. Para definir cada subplot e qual será a posição deles, começamos com a variável `axs` e abrimos colchetes. Nosso primeiro subplot terá a posição `[0,0]`, isto é, a primeira posição na figura.

Logo em seguida, adicionamos a função `plot()` para plotar a informação.

Para facilitar o processo e evitar a criação de DataFrames para cada país, vamos retornar ao DataFrame original, então entre os parênteses da função, digitaremos `df`. Nesse momento, usaremos o método `loc`, utilizado anteriormente para extrair apenas a série de dados do Brasil, e agora para cada um dos países.

Como o primeiro gráfico é referente ao Brasil, passaremos `Brasil` entre colchetes e aspas simples após o método `loc`. Além disso, vamos adicionar a variável `anos` ao método.

```
fig, axs = plt.subplots(2,2, figsize=(10,6))
```

```
axs[0,0].plot(df.loc['Brasil', anos])
```

[COPIAR CÓDIGO](#)

Também podemos definir um título para o subplot para saber do que ele se trata. Usaremos a função `set_title()`, passando para ela `Brasil` entre aspas simples.

```
fig, axs = plt.subplots(2,2, figsize=(10,6))
```

```
axs[0,0].plot(df.loc['Brasil', anos])
```

```
axs[0,0].set_title('Brasil')
```

[COPIAR CÓDIGO](#)

Criando o subplot da Colômbia

Faremos o mesmo processo para os demais países. Trabalharemos agora com a **Colômbia**. Nesse caso, a posição definida será `axs[0,1]`, representando o segundo subplot da figura, ainda na primeira linha.

Entre os parênteses da função `plot()`, vamos colocar novamente `df.loc` e adicionar entre colchetes `Colômbia` e a variável `anos`, para obter o intervalo dos anos.

Para adicionar o título, utilizamos a mesma função `set_title()`, agora recebendo `Colômbia`.

```
fig, axs = plt.subplots(2,2, figsize=(10,6))
```

```
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')
```

[COPIAR CÓDIGO](#)

Criando o subplot da Argentina

O próximo país pode ser **Argentina**. Seguiremos a mesma estrutura de antes, porém com a posição `axs[1,0]`, pois esse subplot será o primeiro da linha de baixo.

```
fig, axs = plt.subplots(2,2, figsize=(10,6))
```

```
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')
```

[COPIAR CÓDIGO](#)

Criando o subplot do Peru

Chegamos ao último país, o **Peru**. Dessa vez, a posição será `axs[1,1]`, correspondente ao segundo subplot da segunda linha.

Por fim, vamos adicionar a função `plt.show()` para visualizar o resultado.

```
fig, axs = plt.subplots(2,2, figsize=(10,6))

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')

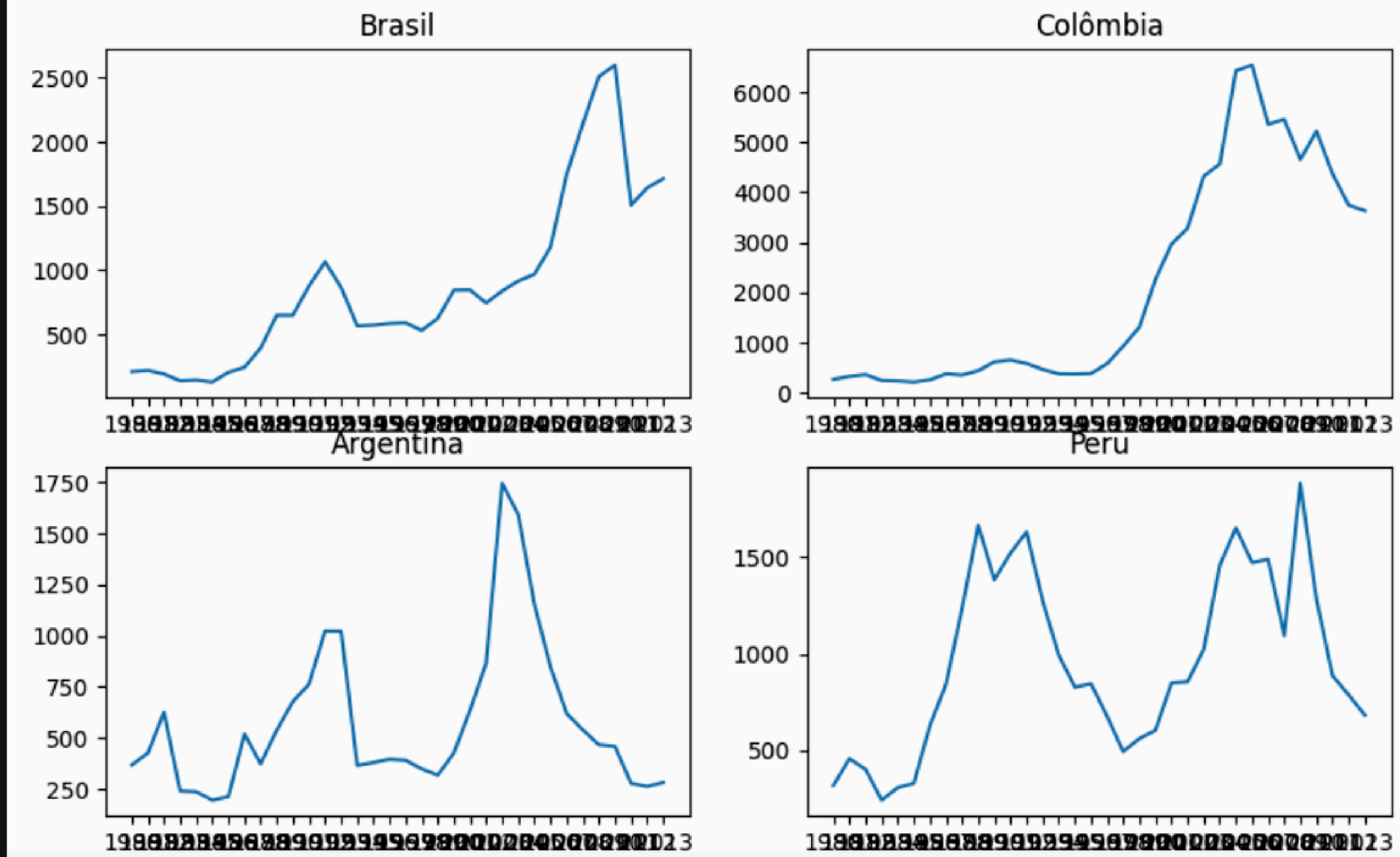
axs[1,0].plot(df.loc['Argentina', anos])
axs[1,0].set_title('Argentina')

axs[1,1].plot(df.loc['Peru', anos])
axs[1,1].set_title('Peru')

plt.show()
```

[COPIAR CÓDIGO](#)

Executando a célula, teremos os seguintes subplots:



Adicionando espaçamento entre os subplots

No momento, a visualização está bastante confusa. Não há uma separação boa entre eles e também não conseguimos entender os valores dos eixos x de cada gráfico.

Vamos começar melhorando o **espaçamento** entre os gráficos. Para isso, vamos adicionar um novo trecho na segunda linha de código, logo abaixo da criação da figura.

Começaremos com a função `fig.subplots_adjust()`, usado para fazer ajustes. Entre os parênteses da função, passaremos o parâmetro `hspace`, responsável por controlar o espaçamento **vertical** entre os subplots de uma figura. Vamos definir o valor de `0.5`.

Em seguida, vamos passar o parâmetro `wspace`, referente ao espaçamento **horizontal**. Nesse caso, vamos testar o valor de `0.3`.

```
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')
```

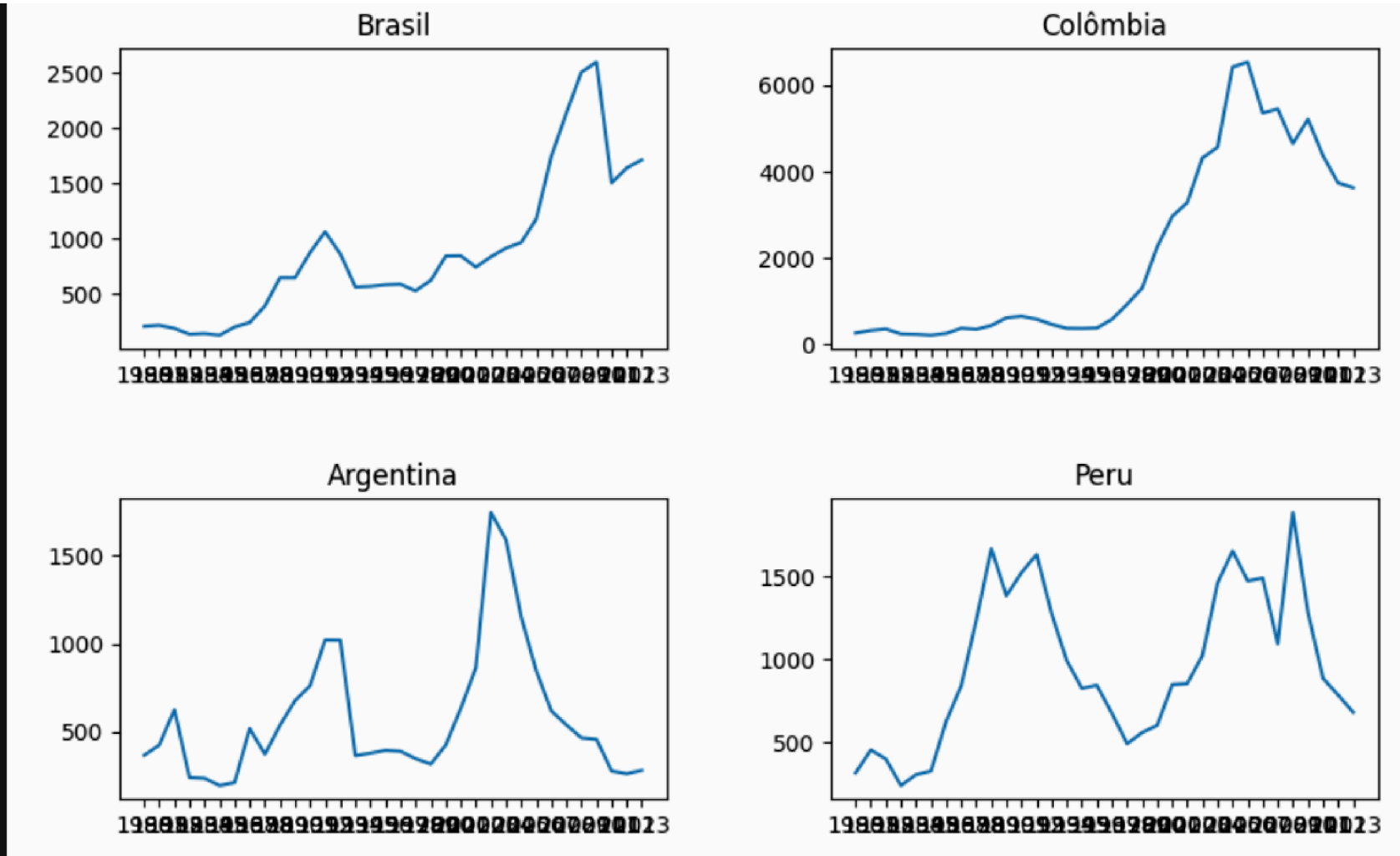


```
axs[1,1].plot(df.loc['Peru', anos])  
  
axs[1,1].set_title('Peru')  
  
plt.show()
```

[COPIAR CÓDIGO](#)

Ajuda da
Luri ✨

[Me explique de outra forma](#)[Me explique utilizando uma analogia](#)[Me dê um exemplo](#)



Ajustando o eixo x dos gráficos

Outro ponto importante é ajustar os valores do eixo x, de modo que seja possível ler e entender os dados do gráfico. Então, vamos usar novamente a frequência a cada 5 anos.

Para aplicar isso a todos os subplots, podemos usar a instrução `for`. Ao final do código, antes da função `plt.show()`, vamos adicionar um novo bloco contendo a instrução `for`, seguida de `ax` (representando o eixo) e da estrutura `in axs`, indicando que serão considerados todos os subplots. Após a variável `axs`, vamos adicionar o atributo `flat` seguido de dois-pontos; ele fará um loop que irá percorrer todos os subplots da matriz.

Na linha de código abaixo, vamos adicionar `ax.xaxis` (para modificar o eixo x) seguido da função `set_major_locator()`. Entre parênteses, colocaremos `plt.MultipleLocator(5)`, para definir a frequência como 5.

```
for ax in axs.flat:  
    ax.xaxis.set_major_locator(plt.MultipleLocator(5))
```

[COPIAR CÓDIGO](#)

Chegaremos ao seguinte resultado da célula:

```
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')

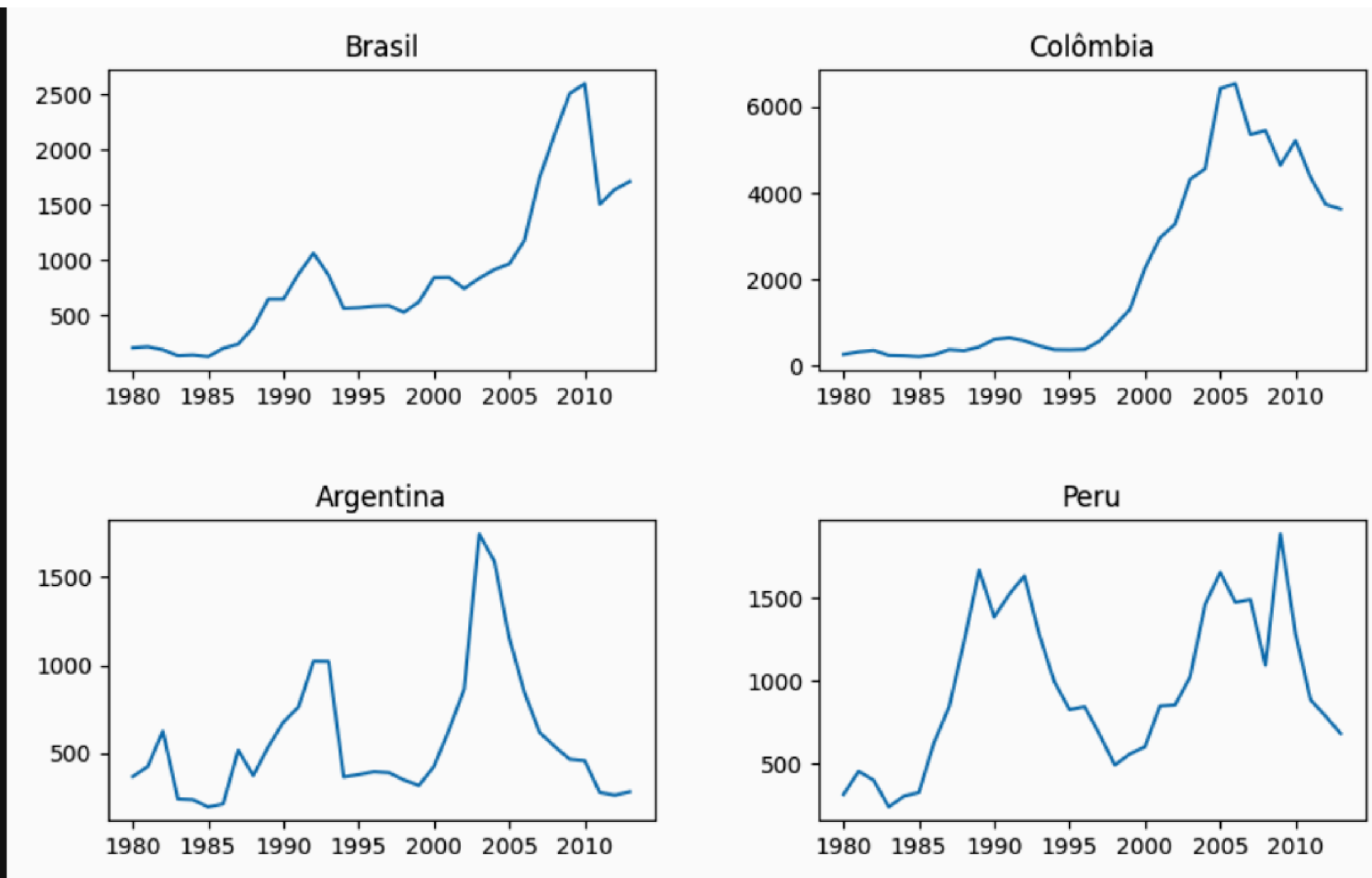
axs[1,1].plot(df.loc['Peru', anos])
axs[1,1].set_title('Peru')

for ax in axs.flat:
    ax.xaxis.set_major_locator(plt.MultipleLocator(5))

plt.show()
```

[COPIAR CÓDIGO](#)

Executando a célula, teremos o seguinte resultado, com o eixo x ajustado na frequência de 5 anos:



Adicionando os rótulos

Nesse momento, podemos adicionar os **rótulos** dos eixos x e y de cada gráfico, usando novamente a instrução `for`. Dessa forma, conseguimos definir os rótulos para todos os gráficos simultaneamente,

em vez de fazer isso para cada subplot individualmente.

Antes da função `plt.show()`, colocaremos um novo bloco com a instrução `for`, seguida novamente de `ax in axs.flat:`.

Na próxima linha de código, vamos adicionar a função `ax.set_xlabel()` (eixo x), passando para ela o `Ano` entre aspas simples. Na linha seguinte, teremos a função `ax.set_ylabel()` (eixo y), recebendo entre parênteses e aspas simples o `Número de imigrantes`.

```
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')

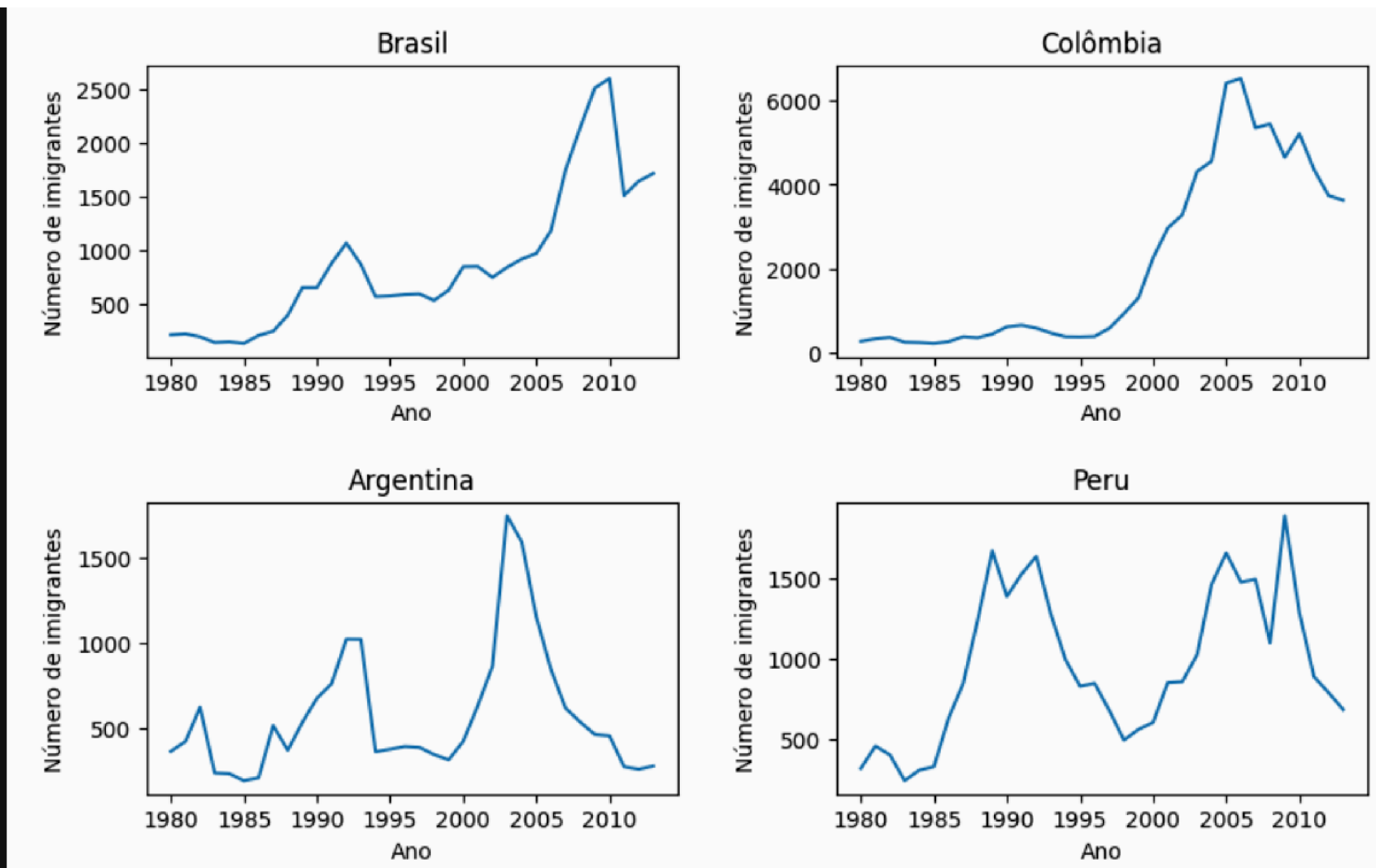
for ax in axs.flat:
    ax.xaxis.set_major_locator(plt.MultipleLocator(5))

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

plt.show()
```

[COPIAR CÓDIGO](#)

Executando a célula, teremos como resultado a figura abaixo:



Conclusão

Agora temos todas as informações importantes nos gráficos. Mas como melhorar ainda mais a nossa figura? **Vamos descobrir no próximo vídeo!**

