



02 Personalizando visualizações com Matplotlib



PRÓXIMA ATIVIDADE



Transcrição

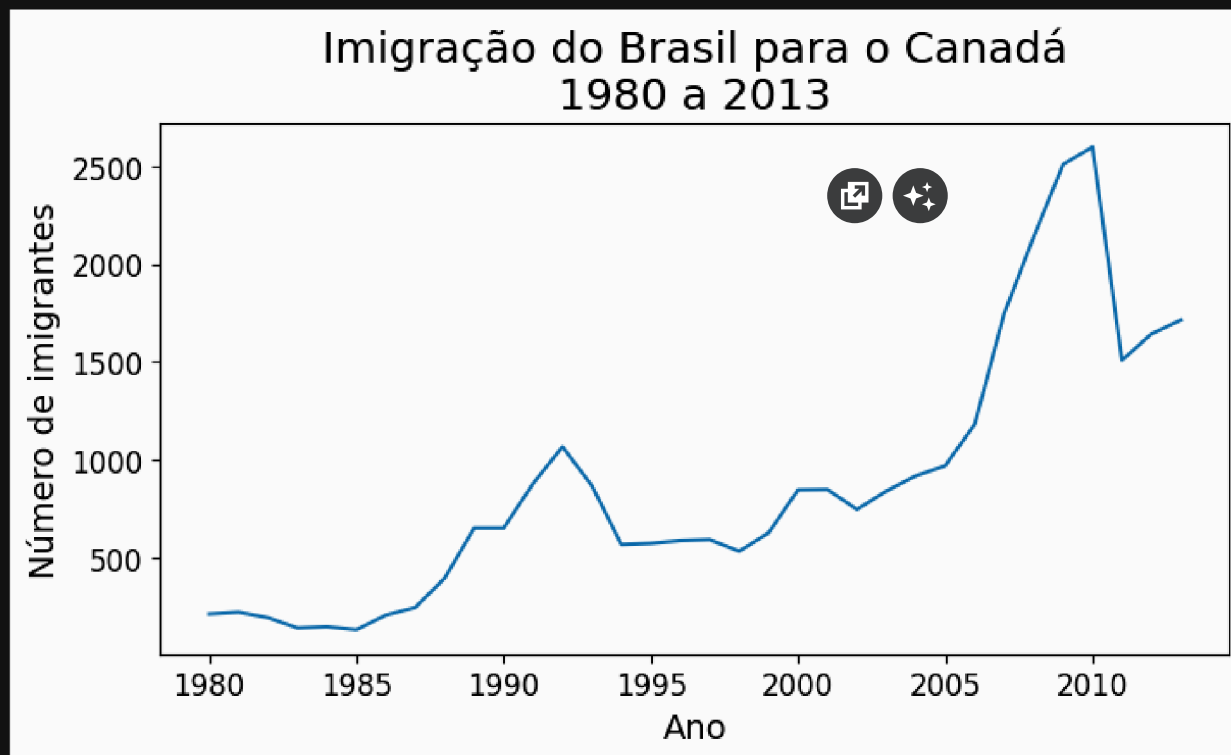
Já aprendemos a plotar gráficos, criar figuras e até a adicionar subplots em uma figura. Agora aprenderemos como customizar ainda mais o nosso gráfico, modificando o tamanho de fontes, mudando as cores, adicionando mais informações.

Para isso, começaremos mudando o gráfico onde tínhamos apenas os dados do Brasil. Estou com o código dessa figura aberto e faremos algumas modificações para melhorar a visualização.

```
fig, ax = plt.subplots(figsize=(8,4))
ax.plot(dados_brasil['ano'], dados_brasil['imigrantes'])
ax.set_title('Imigração do Brasil para o Canadá\n1988 a 2013')
ax.set_xlabel('Ano')
ax.set_ylabel('Número de imigrantes')
ax.xaxis.set_major_locator(plt.MultipleLocator(5))
plt.show()
```

COPIAR CÓDIGO

Gráfico do retorno:



Nossa primeira mudança será o **tamanho da fonte do título**, que costumam ter um tamanho maior. Também modificaremos o tamanho da fonte dos dados numéricos dos eixos X e Y, mas começaremos pelo título.

Para modificarmos o tamanho do título, mudaremos a terceira linha do código, onde temos o `ax.set_title()`. Depois do fechamento de aspas simples do título, escreveremos uma vírgula e usaremos o parâmetro `fontsize`, com o qual conseguimos alterar o tamanho da fonte. Testaremos o valor 18, escrevendo `fontsize=18`.

```
fig, ax = plt.subplots(figsize=(8,4))
ax.plot(dados_brasil['ano'], dados_brasil['imigrantes'])
```

```
ax.set_title('Imigração do Brasil para o Canadá\n1988 a 2013', fontsize=18)
ax.set_xlabel('Ano')
ax.set_ylabel('Número de imigrantes')
ax.xaxis.set_major_locator(plt.MultipleLocator(5))
plt.show()
```

[COPIAR CÓDIGO](#)

Executando a célula, o retorno é o gráfico anterior com o título bem evidente, facilitando a visualização dele no nosso gráfico. Também podemos **alterar a fonte dos rótulos** (*labels*) do eixo X e Y, sucessivamente "Ano" e "Número de imigrantes". Para isso, mudaremos as linhas do `ax.set_xlabel()` e `ax.set_ylabel()`, adicionando o `fontsize=14` depois do nome do rótulo.

```
fig, ax = plt.subplots(figsize=(8,4))
ax.plot(dados_brasil['ano'], dados_brasil['imigrantes'])
ax.set_title('Imigração do Brasil para o Canadá\n1988 a 2013', fontsize=18)
ax.set_xlabel('Ano', fontsize=14)
ax.set_ylabel('Número de imigrantes', fontsize=14)
ax.xaxis.set_major_locator(plt.MultipleLocator(5))
plt.show()
```

[COPIAR CÓDIGO](#)

Ao executarmos o código novamente, o tamanho dos rótulos aumentou também, facilitando a leitura. Nosso próximo passo é **aumentar a fonte dos valores** (*ticks*) do eixo X e do eixo Y.

Essa etapa é um pouco diferente. Para ela, após o fechamento de parênteses da linha `ax.set_ylabel()`, pressionaremos "Enter", e escreveremos novas informações na linha antes do `ax.xaxis.set_major_locator()`.

Começaremos codando `ax.xaxis` para nos referirmos especificamente ao eixo X, e adicionaremos a função `.set_tick_params()`, nos referindo aos parâmetros dos *ticks*. Entre os parênteses, escreveremos `labelsize` para definir o tamanho desse valor.

Como atribuímos "14" para o rótulo, definiremos como "12" para ficar um pouco menor. Sendo assim, o código fica `ax.xaxis.set_tick_params(labelsize=12)`. Pressionaremos "Enter" ao final da linha e seguiremos o mesmo processo para os valores do eixo Y, escrevendo `ax.yaxis.set_tick_params(labelsize=12)`.

```
fig, ax = plt.subplots(figsize=(8,4))
ax.plot(dados_brasil['ano'], dados_brasil['imigrantes'])
ax.set_title('Imigração do Brasil para o Canadá\n1988 a 2013', fontsize=18)
ax.set_xlabel('Ano', fontsize=14)
ax.set_ylabel('Número de imigrantes', fontsize=14)
ax.xaxis.set_tick_params(labelsize=12)
ax.yaxis.set_tick_params(labelsize=12)
ax.xaxis.set_major_locator(plt.MultipleLocator(5))
plt.show()
```

[COPIAR CÓDIGO](#)

Ao executarmos o código, pressionando "Shift + Enter", percebemos que as informações escritas estão mais evidentes, facilitando a leitura do gráfico. Porém, existem mais mudanças que podemos fazer para melhorar a visualização. Uma dessas mudanças é na **posição do título do gráfico**.

No nosso caso, a posição está centralizada, mas a leitura das pessoas costuma acontecer da seguinte forma: em uma página, ela começa do canto superior esquerdo e percorre o olhar em zigue-zague pela página. Então ela lê da esquerda

para direita e volta para o começo da linha abaixo, à esquerda, repetindo esse movimento até chegar ao canto inferior direito da página.

Sendo assim, o ideal é colocarmos o título alinhado à esquerda, porque com isso a pessoa consegue ler o título e acompanhar o gráfico. Essa **não é uma regra**, apenas uma **dica** para melhorar a visualização.

Para mudarmos a posição do título, voltaremos para terceira linha de código, onde temos o `ax.set_title()`. Dentro dos parênteses, após o `fontsize=18`, escreveremos uma vírgula e o parâmetro `loc='left'`, para determinar a localização do título à esquerda.

```
#código omitido  
ax.set_title('Imigração do Brasil para o Canadá\n1988 a 2013', fontsize=18, loc='left')  
#código omitido
```

[COPIAR CÓDIGO](#)

Executando o código, o gráfico recarrega e o título está alinhado à esquerda, algo que percebemos facilmente, porque temos duas linhas de título. Outro aspecto que podemos alterar é a espessura da linha do gráfico, que está bem fina.

Para alterarmos a **espessura de uma linha**, voltamos ao código e alteraremos a linha do `ax.plot()`. Dentro dos parâmetros temos os dados do eixo X e do Y e, depois de `dados_brasil['imigrantes']`, escreveremos uma vírgula seguida do parâmetro `lw=3`, para definirmos a espessura da linha como 3 pixels. Vamos testar se esse tamanho fica agradável.

Ao executarmos o código, pressionando "Shift + Enter", percebemos que a linha está mais grossa e, portanto, mais evidente. Com isso, o gráfico ficou mais interessante e fácil de entender.

Além disso, precisamos **adicionar marcadores na linha**, indicando a posição em que os dados estão. Por exemplo, se temos um dado sobre 1980, a linha teria um marcador nessa posição, e assim por diante.

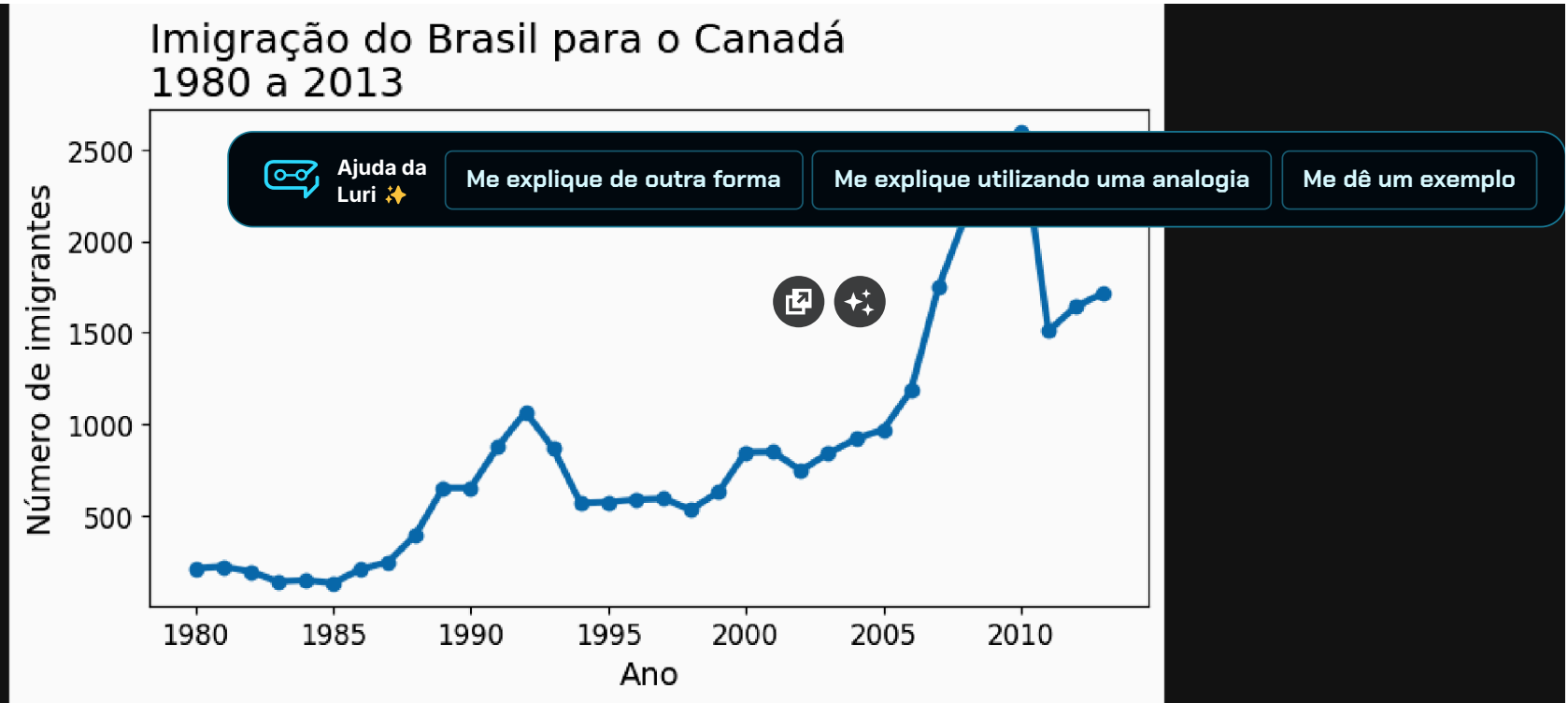
Para isso, voltaremos à linha do `ax.plot()` e, depois do `lw=3`, escreveremos uma vírgula e o parâmetro `marker`. Com esse parâmetro, podemos definir o tipo de marcador que nós queremos: círculo, triângulo, quadrado, entre outras formas. Testaremos com o círculo, codando `marker='o'`.



```
#código omitido  
ax.plot(dados_brasil['ano'], dados_brasil['imigrantes'], lw=3, marker='o')  
#código omitido
```

[COPIAR CÓDIGO](#)

Retorno:



Ao executarmos o código novamente, percebemos que aparecem vários círculos sobre a linha do gráfico, ou seja, temos marcadores em todos os pontos onde temos dados. A adição de marcadores pode ser interessante em alguns casos, mas em outros não.

No nosso caso, temos vários pontos, gerando muita informação, o que sobrecarrega nossa visualização. Sendo assim, não é tão interessante adicionar marcadores para esse gráfico.

Outra coisa que podemos adicionar ao gráfico, e que testamos anteriormente, é **adicionar uma grade (*grid*)** no fundo do gráfico. Então removeremos o `marker='o'` do `ax.plot()`. Em seguida, ao final da penúltima linha do código, antes do `plt.show()`, pressionaremos "Enter". Na nova linha, escreveremos `plt.grid()`.

```
fig, ax = plt.subplots(figsize=(8, 4))
ax.plot(dados_brasil['ano'], dados_brasil['imigrantes'], lw=3)
#código omitido
ax.xaxis.set_major_locator(plt.MultipleLocator(5))
plt.grid()
plt.show()
```

[COPIAR CÓDIGO](#)

Ao executarmos essa célula, pressionando "Shift + Enter", ela retorna o gráfico com um grade cinza claro onde as linhas horizontais correspondem aos *tickers* do eixo Y e as linhas verticais correspondem aos *tickers* do eixo x.

Essa grade serve para orientar onde estão as posições, por exemplo, conseguimos entender que, entre 1990 e 1995, o número de imigrantes está em torno de 1000. Assim fica mais fácil de nos guiarmos e entendermos os valores mostrados ao longo da linha.

Existe também a possibilidade de mudarmos a grade, deixando-a mais suave. Testaremos isso também. Entre os parênteses do `plt.grid()` escreveremos `linestyle='--'`.

```
#código omitido
plt.grid(linestyle='--')
plt.show()
```

[COPIAR CÓDIGO](#)

Ao executarmos a célula e recebermos o gráfico, percebemos que a grade está um pouco diferente. Ao invés de linhas contínuas, elas estão tracejadas, deixando o *grid* mais suavizado.

Com isso, percebemos que uma grade também pode aumentar a densidade visual, e até poluir um pouco a nossa imagem do gráfico. Portanto, é importante pensarmos sempre se adicionar esses elementos deixa as informações mais claras ou atrapalha mais.

Aprendemos como personalizar o gráfico aumentando o tamanho das fontes e dos rótulos ou adicionando marcadores, mas e as cores?

