

```
['Solarize_Light2', '_classic_test_patch', '_mpl-gallery', '_mpl-gallery-nogrid'

COPIAR CÓDIGO
```









Podemos explorar um desses estilos para melhorar ainda mais nossas visualizações!

Antes de aplicar um estilo, é importante saber que cada vez que a biblioteca Matplotlib é importada em um notebook, ela define uma configuração de tempo de execução que inclui os estilos padrão para cada elemento de plotagem criado.

Portanto, para evitar que o estilo seja aplicado a todos os gráficos plotados no mesmo notebook, podemos utilizar um código que cria uma cópia das configurações padrão de plotagem da biblioteca Matplotlib e as atribui à variável IPython_default . Isso pode ser útil para armazenar e reutilizar as configurações padrão de plotagem ou para restaurá-las depois de terem sido modificadas:

Um dos estilos disponíveis é baseado em um site de notícias e análises de dados chamado <u>FiveThirtyEight (https://fivethirtyeight.com/)</u>, que cobre assuntos como política, economia, cultura, ciência e esportes.

Para utilizar o estilo 'fivethirtyeight' usamos o código abaixo:



COPIAR CÓDIGO



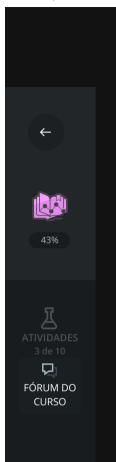


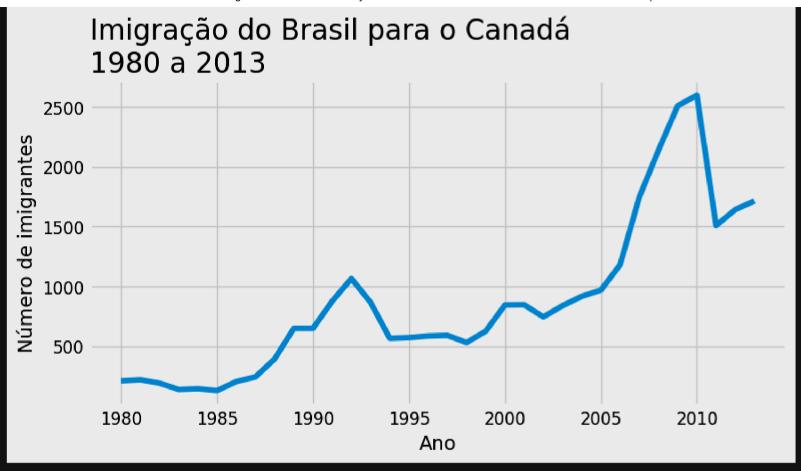


Em seguida, podemos criar a figura, que terá esse novo estilo incorporado:

```
fig, ax = plt.subplots(figsize=(8, 4))
ax.plot(dados_brasil['ano'], dados_brasil['imigrantes'])
ax.set_title('Imigração do Brasil para o Canadá\n1980 a 2013', fontsize=20, loc=
ax.set_ylabel('Número de imigrantes', fontsize=14)
ax.set_xlabel('Ano', fontsize=14)
ax.yaxis.set_tick_params(labelsize=12)
ax.xaxis.set_tick_params(labelsize=12)
ax.xaxis.set_major_locator(plt.MultipleLocator(5))
plt.show()
COPIAR CÓDIGO
```

O resultado será o gráfico exibido abaixo:





Os gráficos criados com o estilo FiveThirtyEight são lindos e possuem estética limpa e minimalista, com linhas mais grossas e cores vibrantes.

Bom, mas e se a gente quiser plotar gráficos sem esse estilo depois de ter definido ele no notebook?

Podemos redefinir os parâmetros utilizando rcParams.update e passando a ele a variável

IPython_default que criamos anteriormente com as configurações padrão:

```
plt.rcParams.update(IPython default);
```

COPIAR CÓDIGO









Além disso, podemos aplicar um estilo apenas a um bloco de código específico, pois o pacote de estilo fornece um gerenciador de contexto para limitar suas alterações a um escopo específico. Para isolar suas alterações de estilo, você pode escrever o código dentro de um contexto with da seguinte forma:

```
with plt.style.context('fivethirtyeight'):
 fig, ax = plt.subplots(figsize=(8, 4))
 ax.plot(dados brasil['ano'], dados brasil['imigrantes'], lw=3)
 ax.set title('Imigração do Brasil para o Canadá\n1980 a 2013', fontsize=20, lo
 ax.set ylabel('Número de imigrantes', fontsize=14)
 ax.set xlabel('Ano', fontsize=14)
 ax.yaxis.set tick params(labelsize=12)
 ax.xaxis.set tick params(labelsize=12)
 ax.xaxis.set major locator(plt.MultipleLocator(5))
 plt.show()
                                                                  COPIAR CÓDIGO
```

Gostou desse novo estilo? A biblioteca Matplotlib oferece uma variedade de estilos, além do FiveThirtyEight, que podemos testar e aplicar aos nossos dados. Com essas opções, podemos tornar nossas visualizações ainda mais atraentes, melhorando a apresentação das informações que queremos transmitir.







