

```
import pandas as pd
import requests
import matplotlib.pyplot as plt
```

```
#consultando dados via api
```

```
URL = "https://economia.awesomeapi.com.br/json/daily/USD-BRL/10"
resposta = requests.get(URL)
print(resposta)
```

```
<Response [200]>
```

```
# Verificando se a requisição foi bem-sucedida
if resposta.status_code == 200:
    dados = resposta.json() # Transforma em lista de dicionários
    df = pd.DataFrame(dados) # Transforma em DataFrame/Tabela

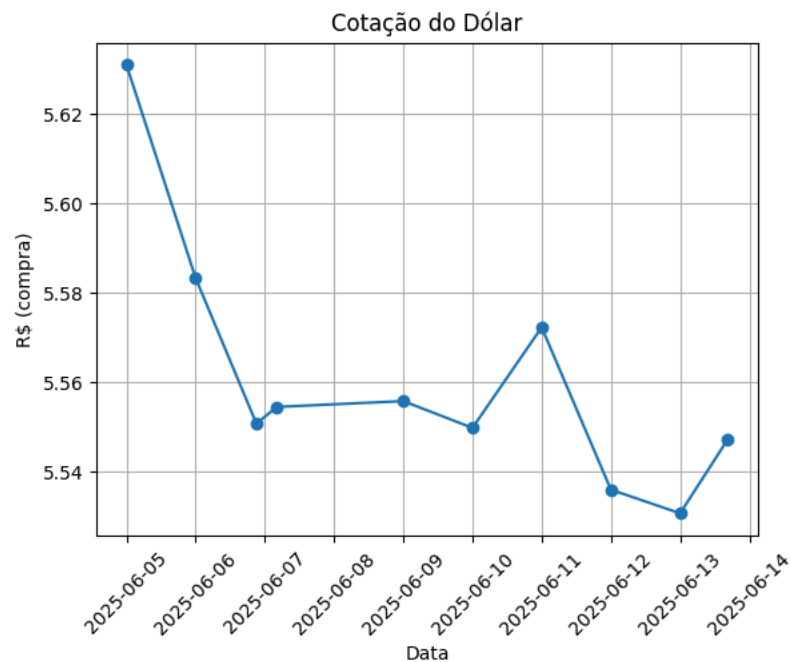
    # Mostrar as primeiras linhas da tabela para conferir os dados
    print(df.head())
    # Convertendo dados para melhor visualização
    df['timestamp'] = pd.to_datetime(df['timestamp'], unit='s') #Retirando o formato de numero inteiro para formato de data
    df['bid'] = df['bid'].astype(float) #mudando o formato para float.
    df['pctChange'] = (df['pctChange'] * 1).round(2)
else:
    print("Erro na requisição:", resposta.status_code) # caso o resultado nao retorne 200, se enquadra como erro na requisição.
```

```
code codein name high low varBid \
0 USD BRL Dólar Americano/Real Brasileiro 5.5925 5.52722 0.0166
1 NaN NaN NaN 5.5576 5.50931 -0.005298
2 NaN NaN NaN 5.5771 5.51845 -0.0363
3 NaN NaN NaN 5.5761 5.53617 0.02249
4 NaN NaN NaN 5.5971 5.54673 -0.00599

pctChange bid ask timestamp create_date
0 0.300147 5.5472 5.5502 1749832517 2025-06-13 13:35:17
1 -0.095705 5.5306 5.54484 1749772773 NaN
2 -0.651445 5.5359 5.5389 1749686338 NaN
3 0.40525 5.5722 5.5752 1749599960 NaN
4 -0.107814 5.54971 5.56382 1749513561 NaN
<ipython-input-77-3057884767>:9: FutureWarning: The behavior of 'to_datetime' with 'unit' when parsing strings is deprecated. In a futur
df['timestamp'] = pd.to_datetime(df['timestamp'], unit='s') #Retirando o formato de numero inteiro para formato de data
```

```
#Criando gráfico de visualização
```

```
plt.plot(df['timestamp'], df['bid'], marker='o') #Timestamp para o eixo X (datas) e bid para o eixo Y (compra). #marker 'o' para adicionar poi
plt.title('Cotação do Dólar')
plt.xlabel('Data')
plt.ylabel('R$ (compra)')
plt.xticks(rotation=45) #criando rotação de 45° para os dados proximos nao fiquem sobrepostos entre si
plt.grid(True) #grade ao fundo do gráfico
plt.show()
```



```
# Gráfico com várias métricas
plt.figure(figsize=(13, 7))
```

```
# Plotando as linhas com legenda para cada uma
plt.plot(df['timestamp'], df['bid'], marker='o', label='Compra (bid)') # Valor de compra do dólar
plt.plot(df['timestamp'], df['ask'], marker='o', label='Venda (ask)') # Valor de venda do dólar
plt.plot(df['timestamp'], df['high'], marker='o', label='Alta diária (high)') # Maior valor do dia
plt.plot(df['timestamp'], df['low'], marker='o', label='Baixa diária (low)') # Menor valor do dia
plt.plot(df['timestamp'], df['pctChange'], marker='x', linestyle='--', label='Variação (%)') # Variação percentual
```

```
# Título e eixos
plt.title('Cotação do Dólar - Últimos Dias (com Variação %)', fontsize=16)
plt.xlabel('Data da Cotação')
plt.ylabel('Valor em R$ e Variação (%)')
```

```
# Girar rótulos do eixo X
plt.xticks(rotation=55)
```

```
# Grade de fundo
plt.grid(True)
```

```
# Legenda explicativa
plt.legend(title='Indicadores mostrados no gráfico', loc='upper right', bbox_to_anchor=(1.35, 1))
```

```
# Mostrar o gráfico
plt.tight_layout()
plt.show()
```

