USAR ESSE CÓDIGO ANTES DE QUALQUER ATIVIDADE -

pip install python-bcb

Seguem 3 diferentes códigos com 3 diferentes formatações dos códigos(Mostrando os gráficos de maneiras diferentes).

import pandas as pd

import matplotlib.pyplot as plt

from bcb import sgs

# === Função para buscar e formatar série ===

def buscar\_serie(codigo, nome\_coluna, data\_inicio):

df = sgs.get({nome\_coluna: codigo}, start=data\_inicio).reset\_index()

df.columns = ['Data', nome\_coluna]

df['Data'] = pd.to\_datetime(df['Data'])

return df

# === Coleta dos dados ===

salario\_minimo = buscar\_serie(1619, 'Salario\_Minimo', '1995-01-01')

ipca = buscar\_serie(433, 'IPCA', '1995-01-01') # Ajustado para mesma base temporal

endividamento = buscar\_serie(19882, 'Endividamento', '1995-01-01')

# === Verificação de valores ausentes antes da limpeza ===

print("Valores ausentes antes do tratamento:")

print(salario\_minimo.isnull().sum())

print(ipca.isnull().sum())

print(endividamento.isnull().sum())

# === Interpolação simples ===

for df in [salario\_minimo, ipca, endividamento]:

df.interpolate(method='linear', inplace=True)

# === Unificação das bases ===

df\_final = pd.merge(salario\_minimo, ipca, on='Data', how='outer')

df\_final = pd.merge(df\_final, endividamento, on='Data', how='outer')

df\_final.sort\_values('Data', inplace=True)

df\_final.interpolate(method='linear', inplace=True)

# === Verificação final ===

print("\nValores ausentes após tratamento:")

print(df\_final.isnull().sum())

# === Exportando arquivos ===

df\_final.to\_csv('Indicadores\_Economicos.csv', index=False)

df\_final.to\_excel('Indicadores\_Economicos.xlsx', index=False)

# === Plot geral ===

plt.figure(figsize=(14, 6))

plt.plot(df\_final['Data'], df\_final['Salario\_Minimo'], label='Salário Mínimo', color='blue')

plt.plot(df\_final['Data'], df\_final['IPCA'] \* 100, label='IPCA x100', color='red', linestyle='dashed')

plt.plot(df\_final['Data'], df\_final['Endividamento']\* 10, label='Endividamento (%)', color='green')

plt.xlabel('Ano')

plt.ylabel('Valores')

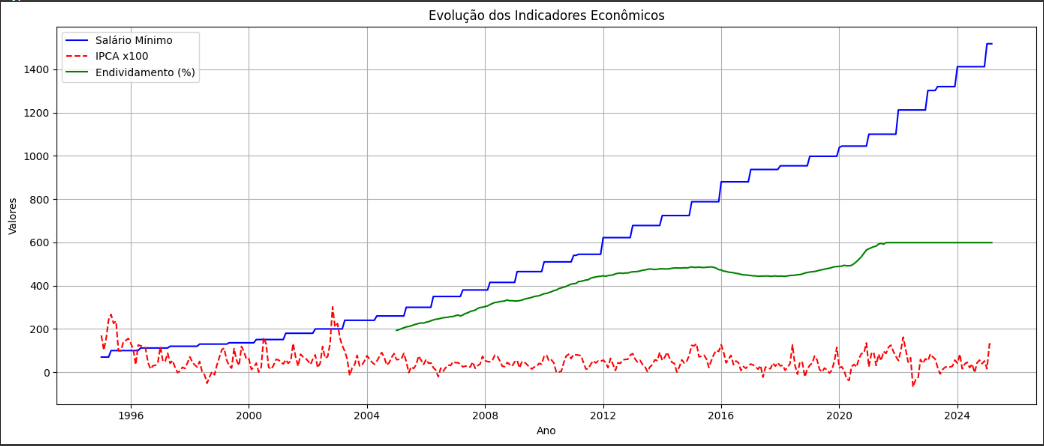
plt.title('Evolução dos Indicadores Econômicos')

plt.legend()

plt.grid(True)

plt.tight\_layout()

plt.show()



Segundo código -

import pandas as pd

from bcb import sgs

import matplotlib.pyplot as plt

# =======================

# 2. Coleta dos Dados

# =======================

# Salário Mínimo

salario\_minimo = sgs.get({'salario\_minimo': 1619}, start='1995-01-01').reset\_index()

salario\_minimo.columns = ['data', 'salario\_minimo']

# IPCA

ipca = sgs.get({'ipca': 433}, start='1995-01-01').reset\_index()

ipca.columns = ['data', 'ipca']

# Endividamento

endividamento = sgs.get({'endividamento': 19882}, start='1995-01-01').reset\_index()

endividamento.columns = ['data', 'endividamento']

# =======================

# 3. Limpeza

# =======================

# Garantir formato de data

for df in [salario\_minimo, ipca, endividamento]:

df['data'] = pd.to\_datetime(df['data'])

df.dropna(inplace=True)

# =======================

# 4. Integração

# =======================

df\_geral = pd.merge(salario\_minimo, ipca, on='data', how='inner')

df\_geral = pd.merge(df\_geral, endividamento, on='data', how='inner')

# =======================

# 5. Transformações

# =======================

# Crescimento percentual acumulado do salário mínimo

valor\_inicial = df\_geral['salario\_minimo'].iloc[0]

df\_geral['salario\_minimo\_percentual'] = (df\_geral['salario\_minimo'] / valor\_inicial - 1) \* 10

# Média móvel do IPCA (opcional)

df\_geral['ipca\_mm12'] = df\_geral['ipca'].rolling(window=12).mean()

# =======================

# 6. Visualização

# =======================

plt.figure(figsize=(14,6))

plt.plot(df\_geral['data'], df\_geral['salario\_minimo\_percentual'], label='Salário Mínimo (% Acumulado)')

plt.plot(df\_geral['data'], df\_geral['ipca'], label='IPCA (%)')

plt.plot(df\_geral['data'], df\_geral['endividamento'], label='Endividamento (%)')

plt.title('Comparação: Salário Mínimo x IPCA x Endividamento')

plt.xlabel('Ano')

plt.ylabel('Percentual (%)')

plt.grid(True)

plt.ylim(0, 100) # Intervalo de 0 a 100 para facilitar a leitura

plt.yticks(range(0, 101, 10)) # De 10 em 10

plt.legend()

plt.tight\_layout()

plt.show()

# =======================

# 7. Exportação

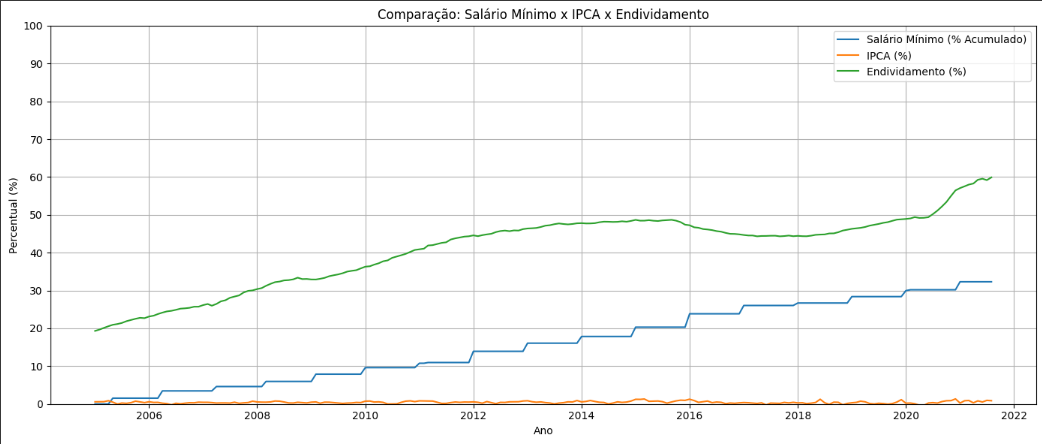
# =======================

# Seleciona as colunas finais

df\_final = df\_geral[['data', 'salario\_minimo\_percentual', 'ipca', 'ipca\_mm12', 'endividamento']]

df\_final.to\_csv('dados\_transformados.csv', index=False)

df\_final.to\_excel('dados\_transformados.xlsx', index=False)



Código 3

import pandas as pd

from bcb import sgs

import matplotlib.pyplot as plt

# Coleta

salario\_minimo = sgs.get({'salario\_minimo': 1619}, start='2005-01-01').reset\_index()

salario\_minimo.columns = ['data', 'salario\_minimo']

salario\_minimo = salario\_minimo.set\_index('data').resample('MS').ffill().reset\_index()

ipca = sgs.get({'ipca': 433}, start='2005-01-01').reset\_index()

ipca.columns = ['data', 'ipca']

ipca = ipca.set\_index('data').resample('MS').ffill().reset\_index()

endividamento = sgs.get({'endividamento': 19882}, start='2005-01-01').reset\_index()

endividamento.columns = ['data', 'endividamento']

endividamento = endividamento.set\_index('data').resample('MS').ffill().reset\_index()

# Merge dos dados

df\_geral = pd.merge(salario\_minimo, ipca, on='data', how='inner')

df\_geral = pd.merge(df\_geral, endividamento, on='data', how='inner')

# Cálculo do crescimento percentual acumulado

valor\_inicial = df\_geral['salario\_minimo'].iloc[0]

df\_geral['salario\_minimo\_percentual'] = (df\_geral['salario\_minimo'] / valor\_inicial - 1) \* 10

# Média móvel do IPCA

df\_geral['ipca\_mm12'] = df\_geral['ipca'].rolling(window=12).mean()

# Visualização

fig, ax1 = plt.subplots(figsize=(14,6))

ax1.plot(df\_geral['data'], df\_geral['salario\_minimo\_percentual'], label='Salário Mínimo (% Acumulado)', color='blue')

ax1.plot(df\_geral['data'], df\_geral['endividamento'], label='Endividamento (%)', color='green')

ax1.set\_ylabel('Percentual (%)')

ax1.set\_ylim(0, 100)

ax1.set\_yticks(range(0, 101, 10))

ax1.legend(loc='upper left')

ax1.grid(True)

ax2 = ax1.twinx()

ax2.plot(df\_geral['data'], df\_geral['ipca'], label='IPCA (%)', color='orange')

ax2.set\_ylabel('IPCA (%)')

ax2.legend(loc='upper right')

plt.title('Comparação: Salário Mínimo x IPCA x Endividamento')

plt.xlabel('Ano')

plt.tight\_layout()

plt.show()

