

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())

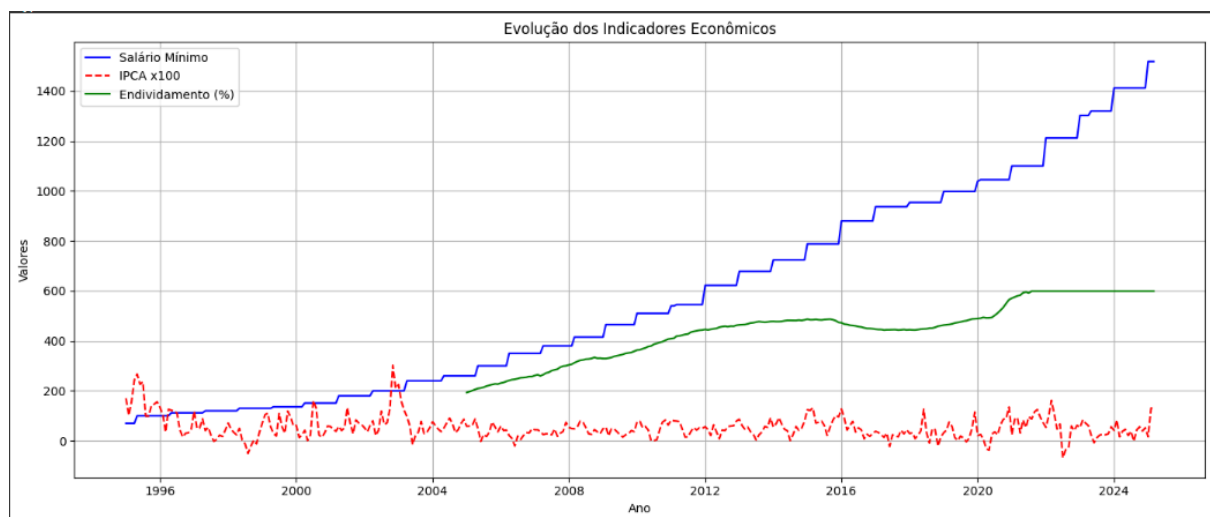
# === Interpolação simples ===
for df in [salario_minimo, ipca, endividamento]:
    df.interpolate(method='linear', inplace=True)

# === 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
# =====

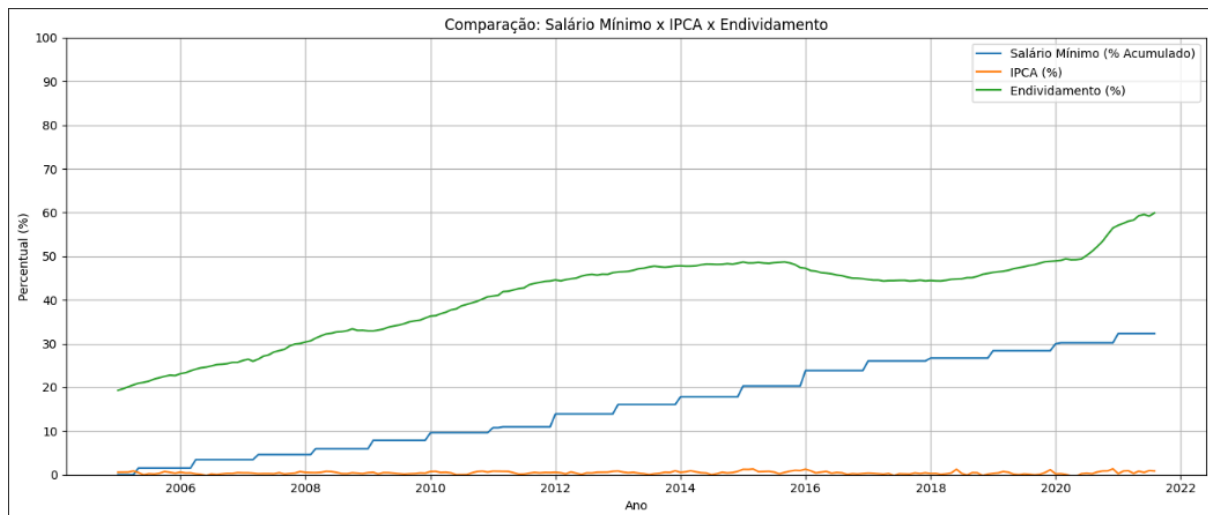
plt.figure(figsize=(14,6))
plt.plot(df_geral['data'], df_geral['salario_minimo_percentual'],
label='Salário Mínimo (% Acumulado)')
plt.plot(df_geral['data'], df_geral['ipca'], label='IPCA (%)')
plt.plot(df_geral['data'], df_geral['endividamento'],
label='Endividamento (%)')
plt.title('Comparação: Salário 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()
```

