

Gerando o Gráfico

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
import numpy as np
import os

def realizar_regressao_mensal(csv_path: str):
    """
    Realiza a regressão linear entre Consumo (kW) e Temperatura Média (C)
    em um nível de agregação mensal para melhorar a correlação R.

    @param csv_path: Caminho para o arquivo CSV unificado.
    @return: Tuple contendo o coeficiente angular (slope), o intercepto, R2 e R.
    """

    # === 1. Carregar e Pré-processar o CSV ===
    df = pd.read_csv(csv_path)

    # Seleciona apenas as colunas necessárias e remove NaNs
    df = df[["data", "consumo_kw", "temperatura_media"]].dropna()
    df["data"] = pd.to_datetime(df["data"], errors="coerce")

    # === PONTO DE CORREÇÃO: Agregação Mensal ===
    # Agrupa por Ano e Mês, calculando a MÉDIA de Consumo e Temperatura
    df_mensal = df.set_index("data").resample("M").mean().dropna()

    # Se a agregação resultar em menos de 3 pontos, a regressão será fraca/impossível
    if len(df_mensal) < 3:
        print("[AVISO] Menos de 3 pontos de dados mensais. Regressão pode ser inválida.")

    # === 2. Preparar os dados (Mensais) ===
    # Reseta o índice para usar a temperatura_media como variável independente
    X = df_mensal["temperatura_media"].values.reshape(-1, 1) # Variável independente
    (Mensal)
    y = df_mensal["consumo_kw"].values # Variável dependente (Mensal)

    # === 3. Criar e treinar o modelo de regressão linear ===
    model = LinearRegression()
    model.fit(X, y)

    # === 4. Previsões para a linha de regressão ===
    y_pred = model.predict(X)

    # === 5. Calcular R2 e R ===
```

```

r2 = r2_score(y, y_pred)
r = np.sqrt(r2) * np.sign(model.coef_[0]) # R com sinal do coeficiente

# === 6. Criar diretório para salvar gráficos ===
os.makedirs("./graficos", exist_ok=True)

# === 7. Plotar os dados e a linha de regressão ===
plt.figure(figsize=(10, 6))

# Plota os dados mensais (agregados)
plt.scatter(X, y, color='blue', label="Dados Reais (Média Mensal)", alpha=0.7, s=100)

# Plota a linha de regressão
plt.plot(X, y_pred, color='red', linewidth=2, label="Regressão Linear Mensal")

plt.xlabel("Temperatura Média Mensal (°C)")
plt.ylabel("Consumo Médio Mensal (kW)")
plt.title("Regressão Linear MENSAL: Consumo x Temperatura Média")
plt.legend()
plt.grid(True)

# Adicionar R no gráfico
plt.text(
    0.05, 0.95,
    f"R = {r:.3f}\nR² = {r2:.3f}\nN={len(df_mensal)} meses",
    transform=plt.gca().transAxes,
    fontsize=12,
    verticalalignment='top',
    bbox=dict(boxstyle="round,pad=0.3", facecolor="white", edgecolor="black")
)

# === 8. Salvar o gráfico ===
file_name = f"regressao_consumo_temperatura_mensal_{len(df_mensal)}.png"
plt.savefig(f"./graficos/{file_name}", dpi=300)
plt.show()

# === 9. Coeficientes da regressão ===
print("--- Resultados da Regressão Linear MENSAL ---")
print(f"Coeficiente angular (slope) - Beta 1: {model.coef_[0]:.4f}")
print(f"Intercepto - Beta 0: {model.intercept_[0]:.4f}")
print(f"R²: {r2:.4f}")
print(f"R: {r:.4f}")

return model.coef_[0], model.intercept_[0], r2, r

# === Execução Principal ===
if __name__ == "__main__":
    realizar_regressao_mensal("dados_unificados.csv")

```

