



IPRJ/UERJ - INSTITUTO POLITÉCNICO DO ESTADO DO RIO DE JANEIRO

LUCAS RODRIGUES ESTORCK PINTO

**OTIMIZAÇÃO:** Previsão dos preços de uma ação altamente volátil

NOVA FRIBURGO  
2024

LUCAS RODRIGUES ESTORCK PINTO

**OTIMIZAÇÃO:** Previsão dos preços de uma ação altamente volátil

Trabalho apresentado ao curso de Engenharia da Computação  
na disciplina Modelagem Computacional com o intuito de  
analisar a precisão de um modelo de regressão polinomial

Professor responsável: Gustavo Libotte

NOVA FRIBURGO  
2024

## SUMÁRIO

INTRODUÇÃO.....	3
O MÉTODO.....	4
RESULTADOS OBTIDOS.....	5
RESÍDUOS ACUMULADOS.....	8
LUCROS/PREJUÍZOS.....	10
EXPLICAÇÃO DO CÓDIGO.....	12
CONCLUSÃO.....	15

## INTRODUÇÃO

Neste trabalho, será visto o método de otimização dos mínimos quadrados para tentar prever o preço da ação analisada após um horizonte de 15 dias após a compra.

Na situação problema, um investidor compra um montante de 10.000 (dez mil) papéis de PETR4 e, após 15 dias irá eminentemente, sem nenhum tipo de análise, vender as suas ações, o objetivo será tentar analisar, através de polinômios de grau 6 (seis), 7 (sete) e 8 (oito), o possível lucro/prejuízo do investimento, ou melhor, especulação. Lembre-se, que será usado um range de 3 meses para coleta de dados, abrindo mão de grande parte do data frame fornecido para este trabalho.

Para contexto, a ação PETR4 se refere ao papel preferencial da PETROBRAS, empresa que sua maior fatia acionária é o Governo Federal, caracterizando-a como uma estatal. Por esse fator, pelo fator fiscal do Brasil e também pelo fato da flutuação dos valores do petróleo, definidos majoritariamente pelo oriente médio, os papéis da PETROBRAS caracterizam grande volatilidade e risco, fazendo assim com que o erro associado aos resultados que aqui serão obtidos, sejam enormes, portanto, vale ressaltar que de maneira alguma esses métodos podem ser usados como única ferramenta para especulação financeira.

## O MÉTODO

O método de regressão polinomial por mínimos quadrados funciona como uma extensão do método linear, esse método nos permite ajustar um polinômio de grau  $n$  em relação aos dados fornecidos, portanto, o método consiste em encontrar os coeficientes que compõem o polinômio em questão.

$$y = (\beta_0) + (\beta_1)x + (\beta_2)x^2 + \dots + (\beta_n)x^n$$

Acima, observamos a forma geral do polinômio que será encontrado, vale ressaltar que a escolha do grau do polinômio é muito importante, pois graus pequenos, podem ter precisão insuficiente, assim como graus muito elevados, podem trazer um excesso de precisão desnecessário, desperdiçando poder computacional.

Portanto, o objetivo aqui é encontrar os coeficientes  $\beta_0, \beta_1, \dots, \beta_n$ , abrindo mão de alguns detalhes, pois o intuito deste documento não é esse, chegamos a conclusão que essa minimização da soma dos quadrados nos fornece um sistema de equações algébricas, como dispõe abaixo.

$$\begin{bmatrix} n+1 & \sum_{i=0}^n x_i & \dots & \sum_{i=0}^n x_i^m \\ \sum_{i=0}^n x_i & \sum_{i=0}^n x_i^2 & \dots & \sum_{i=0}^n x_i^{m+1} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=0}^n x_i^m & \sum_{i=0}^n x_i^{m+1} & \dots & \sum_{i=0}^n x_i^{2m} \end{bmatrix} \begin{bmatrix} a_0 \\ \vdots \\ a_m \end{bmatrix} = \begin{bmatrix} \sum_{i=0}^n f_i \\ \vdots \\ \sum_{i=0}^n x_i^m f_i \end{bmatrix}$$

Fonte: <https://www.math.tecnico.ulisboa.pt/~calves/cursos/mmq.htm>

Não devemos esquecer também os resíduos acumulados, que se dão pela seguinte notação:

$$\varepsilon_i = y_i - \hat{y}_i$$

Onde  $\hat{y}_i$  é o y ajustado no polinômio encontrado.

Dado as informações acima, portanto temos o residuo acuulado até i(ponto):

$$\sum \varepsilon_i .$$

## RESULTADOS OBTIDOS

- Os coeficientes encontrados para o polinômio de grau 6 (a) foram:

```
Coeficientes do polinômio de grau 6:
[ 5.04364122e-09 -9.03253717e-07  5.76624254e-05 -1.89962751e-03
 4.98308284e-02 -8.76952089e-01  4.01829424e+01]
```

- Os coeficientes encontrados para o polinômio de grau 7 (b) foram:

```
Coeficientes do polinômio de grau 7:
[ 1.62382132e-09 -3.47325586e-07  2.92737169e-05 -1.23349974e-03
 2.68303126e-02 -2.62189294e-01  4.66860377e-01  3.90101892e+01]
```

- Os coeficientes encontrados para o polinômio de grau 8 (c) foram:

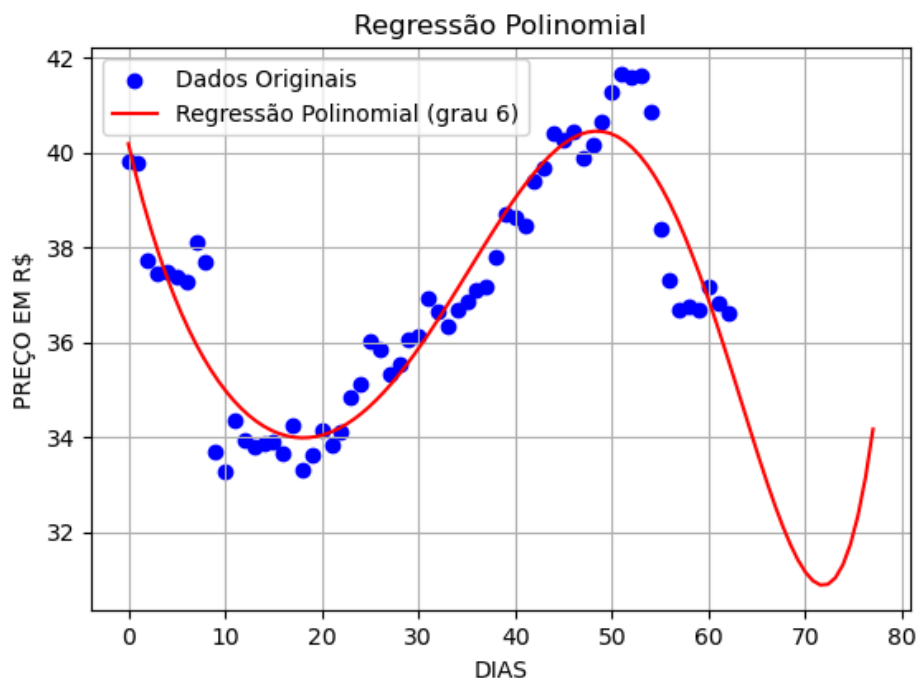
```
Coeficientes do polinômio de grau 8:
[ 5.20600238e-11 -1.12870646e-08  9.57541556e-07 -3.97285128e-05
 8.03329163e-04 -6.23370589e-03  7.45897477e-03 -4.12083630e-01
 3.95616367e+01]
```

Observe, que, um grau maior para o polinômio, gera um polinômio que se comporta muito bem passando pelos pontos já disponíveis, porém, pode ser que não se comporte tão bem na continuação do polinômio, não respeitando o “padrão” dos pontos já conhecidos, esse fenômeno se chama *overfitting*, de mesma maneira ocorre quando o grau do polinômio é muito baixo, resultando em um *underfitting*, subestimando a complexidade associada à situação problema. Para sabermos se esse

é o caso dado os graus que foram escolhidos, podemos analisar os gráficos dos polinômios gerados.

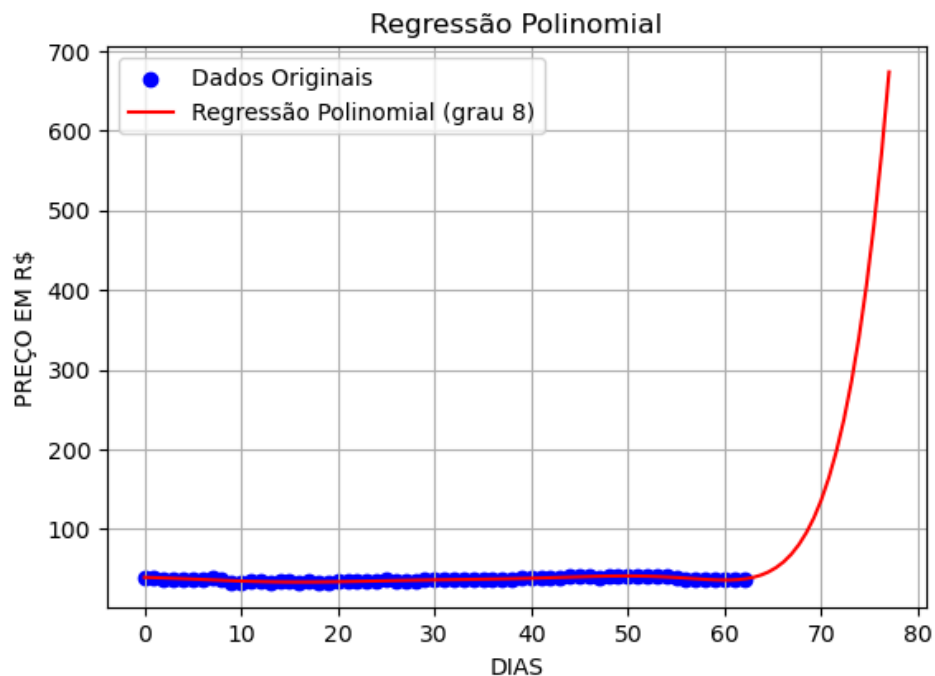
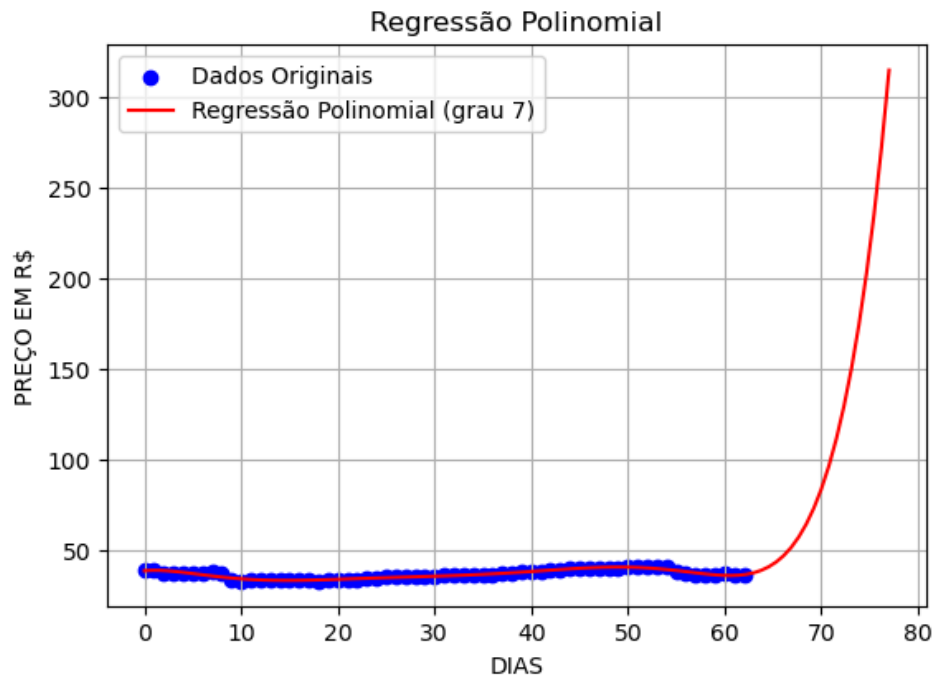
Sendo assim, nossa grande preocupação aqui é também ser crítico o suficiente para saber abrir mão de uma “grande precisão” para ter resultados condizentes com a realidade e, obviamente, que sigam um rigor de precisão.

- **Gráfico do polinômio de grau 6:**



Observe no gráfico acima que o polinômio de grau 6 parece se comportar de maneira satisfatória, estando próximo dos valores reais que foram fornecidos.

- Gráfico do polinômio de grau 7 e 8:



Dados os gráficos dos polinômios acima em relação aos pontos dados, vemos um caso claro de **overfitting**, no qual os graus 7 e 8 representam muito bem os dados disponíveis porém extrapolam no momento em que não há mais dados, fazendo com



que tentar fazer previsões com um modelo em **overfitting**, faça com que os resultados sejam completamente distantes do que se espera, e, acima disso, sejam irreais.

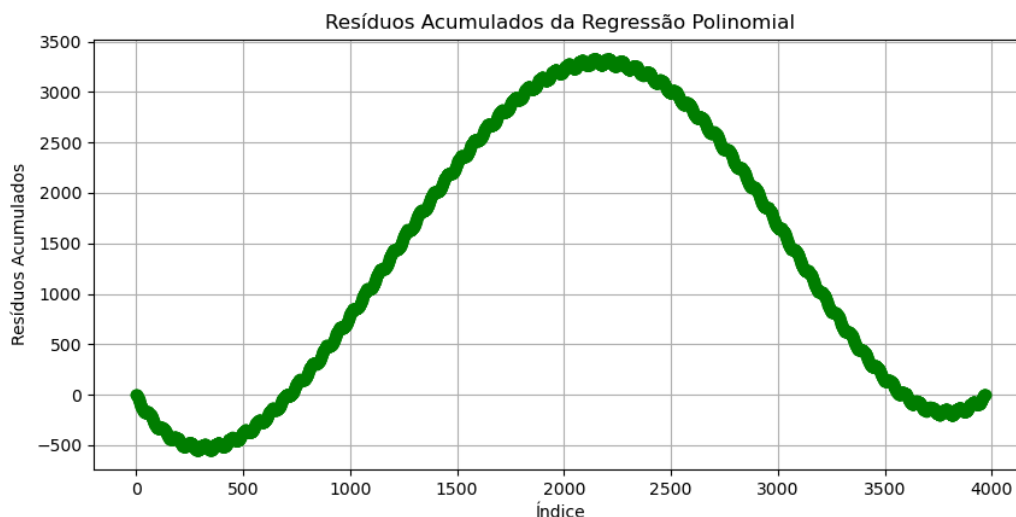
Portanto, a análise dos graus dos polinômios e seus respectivos gráficos, nos ajudou a concluir que, para o investidor seria muito mais interessante olhar apenas para o polinômio de grau 6, porém para fins didáticos, nós trataremos os 3 graus dos polinômios neste documento.

## RESÍDUOS ACUMULADOS

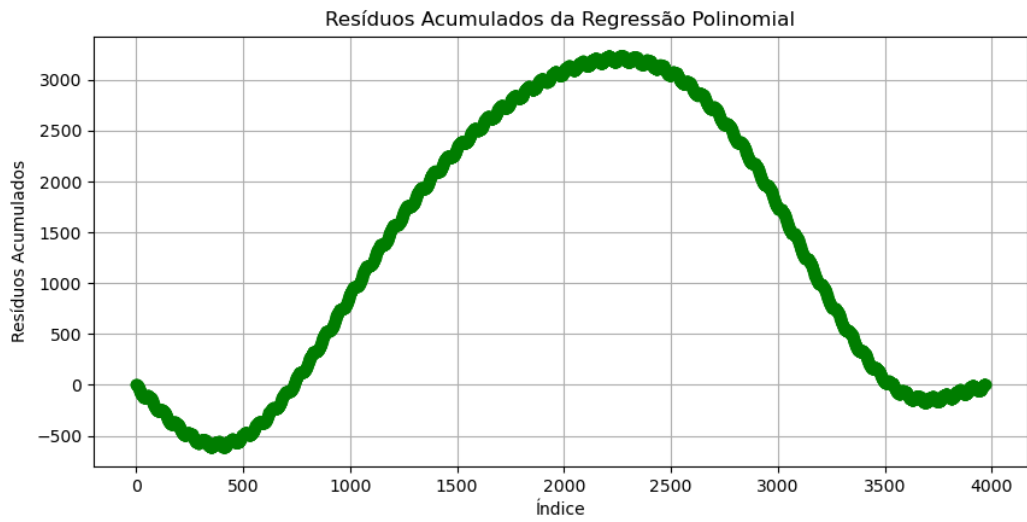
Como vimos algumas seções atrás, o resíduo acumulado se dá pelo somatório de  $\varepsilon_i$ , que por sua vez, é igual a:

$$\varepsilon_i = y_i - \hat{y}_i$$

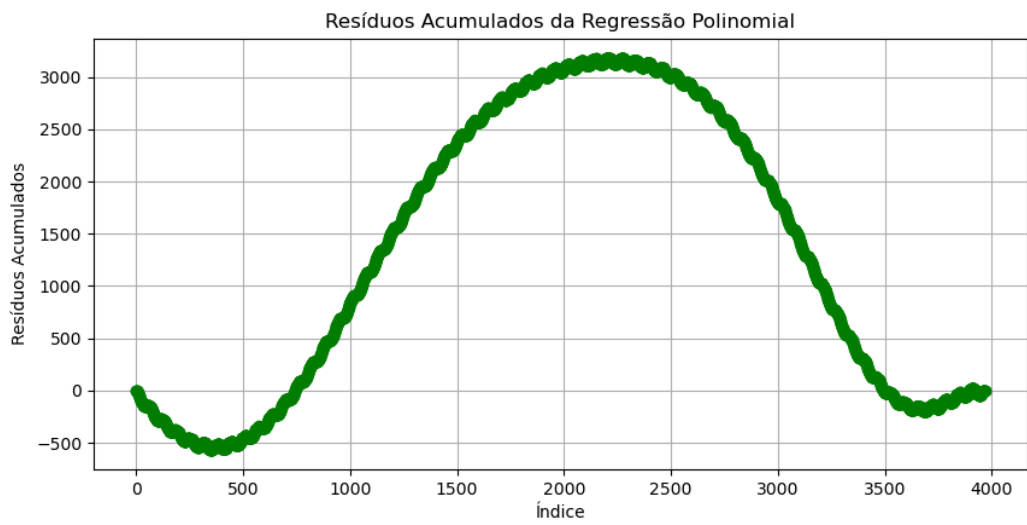
Dito isso, aqui está os gráficos dos resíduos acumulados para cada grau de polinômio:



Resíduo acumulado para o polinômio de grau 6.



Resíduo acumulado para o polinômio de grau 7.



Resíduo acumulado para o polinômio de grau 8.

## LUCROS/PREJUÍZOS

Antes de iniciar as análises, vamos inferir que o investidor comprou 10.000 papéis no dia 22/02/2024, com um preço médio de R\$39,36, tendo um custo total de R\$396.600,00. Portanto, agora podemos analisar cada um dos casos.

### - Caso para o polinômio ajustado de grau 6:

```
Previsão de preço da ação para os próximos 15 dias:  
Dia 1: [34.9954129]  
Dia 2: [34.33918476]  
Dia 3: [33.69043875]  
Dia 4: [33.06399554]  
Dia 5: [32.47668628]  
Dia 6: [31.94748212]  
Dia 7: [31.49762725]  
Dia 8: [31.15077568]  
Dia 9: [30.93313156]  
Dia 10: [30.87359322]  
Dia 11: [31.00390077]  
Dia 12: [31.35878732]  
Dia 13: [31.97613395]  
Dia 14: [32.89712815]  
Dia 15: [34.16642603]
```

Observe que no dia 15, se passaram cerca de 3 meses e 15 dias após a compra, caso o investidor venda seus 10.000 papéis de PETR4 ao preço de R\$34.17 por unidade, o valor total da venda será de R\$341.700,00, caracterizando um **prejuízo** de R\$54.900.

### - Caso para o polinômio ajustado de grau 7:

Neste caso, antes de começar a análise, vale ressaltar que a partir do grau 7, como mostrado na seção **Resultados Obtidos**, temos um overfitting, fazendo com que as previsões a partir daqui sejam extremamente exageradas e, obviamente, não confiáveis.

```
Previsão de preço da ação para os próximos 15 dias:  
Dia 1: [37.85405031]  
Dia 2: [39.69912991]  
Dia 3: [42.59069639]  
Dia 4: [46.80594535]  
Dia 5: [52.66715939]  
Dia 6: [60.54639857]  
Dia 7: [70.87048103]  
Dia 8: [84.12626184]  
Dia 9: [100.86621824]  
Dia 10: [121.71434957]  
Dia 11: [147.37239999]  
Dia 12: [178.62641217]  
Dia 13: [216.35362019]  
Dia 14: [261.52968978]  
Dia 15: [315.23631412]
```

Aqui, o investidor, aproximadamente 3 meses e 15 dias após a compra dos 10.000 papéis de PETR4 com o custo de aquisição de R\$396.600,00, vendeu cada papel com um extraordinário preço de R\$315,24, resultando em um valor total de venda de R\$3.152.400,00, dando um **lucro** ao investidor de R\$2.782.800,00.

- **Caso para para o polinômio ajustado de grau 8:**

```
Previsão de preço da ação para os próximos 15 dias:  
Dia 1: [39.58926125]  
Dia 2: [43.38645315]  
Dia 3: [49.28126932]  
Dia 4: [57.90128452]  
Dia 5: [69.99760352]  
Dia 6: [86.4615489]  
Dia 7: [108.34285584]  
Dia 8: [136.86945689]  
Dia 9: [173.4689414]  
Dia 10: [219.79177648]  
Dia 11: [277.73637864]  
Dia 12: [349.47612698]  
Dia 13: [437.48841133]  
Dia 14: [544.58581053]  
Dia 15: [673.94949828]
```

Sabemos que o custo total de compra dos 10.000 papéis de PETR4 foi de R\$396.600,00. Observe que no 15º dia, as ações da PETROBRAS possuem um valor

de R\$673,95, cerca de duas vezes mais que no polinômio ajustado anterior. Sendo assim, o valor total da venda foi de R\$6.739.500,00, resultando em um lucro exorbitante de R\$6.342.900,00.

## EXPLICAÇÃO DO CÓDIGO

Nesta seção, veremos a explicação do algoritmo que foi desenvolvido para obter os resultados mostrados neste documento. Vale ressaltar que a linguagem de programação utilizada foi o python, em sua versão bleeding edge do Arch Linux, sendo a versão 3.12.4 no dia 21/06/2024. O código foi construído de maneira com que sua leitura seja fácil, seguindo as regras de formatação da PEP 8. O editor de texto usado para construir o código foi o Atom, na versão estável 1.63.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

Primeiramente, foram importadas as bibliotecas acima, para manipulação numérica e plotagem dos gráficos. A denotação “as” é utilizada para dar um “apelido” para a biblioteca, facilitando suas chamadas.

```
# Carregar os dados do CSV
caminho = 'petr4.csv'
dados = pd.read_csv(caminho, sep=';', decimal=',')

# Converter a coluna 'DateTime' para datetime
dados['DateTime'] = pd.to_datetime(dados['DateTime'], format='%Y/%m/%d')

# Filtrar os dados dos últimos 3 meses
data_fim = dados['DateTime'].max()
data_inicio = data_fim - pd.DateOffset(months=3)
dados_filtrados = dados[(dados['DateTime'] >= data_inicio) & (dados['DateTime'] <= data_fim)]
```

Primeiramente, definimos uma variável com o caminho do arquivo CSV para então criar uma variável para leitura dos dados nela contidos. Observe que o separador e o decimal foram definidos, para evitar problemas de leitura e de uso.

Após isso, foi configurado o DateTime no formato ano/mês/dia, como foi observado no arquivo CSV, para que fosse possível definir o range de 3 meses, como foi solicitado no enunciado do trabalho.

```
# Preparar os dados
x = np.arange(len(dados_filtrados)).reshape(-1, 1) # Usar os índices como valores de x
y = dados_filtrados['Series 1'].values

print(dados_filtrados.head())
```

Nesta passagem, foram definidos os eixos x e y, e após isso foi printado os dados filtrados como uma maneira de debug, para confirmação de que tudo foi definido de maneira coerente.

```
def regressao_polinomial(x, y, grau):

    x = np.array(x).flatten()
    y = np.array(y)

    # Ajustar o polinômio usando numpy's polyfit
    coeficientes = np.polyfit(x, y, grau)

    return coeficientes
```

Na função acima, usamos o método do numpy polyfit que gera os coeficientes dos polinômios e retorna em uma lista.

```
def plot_regressao_polinomial(x, y, coeficientes):

    plt.scatter(x, y, color='blue', label='Dados Originais')

    # Calcular os valores ajustados
    x_fit = np.linspace(min(x), max(x) + 15, 100) # Estender para 15 dias além dos dados originais
    y_fit = np.polyval(coeficientes, x_fit)

    # Plotar a linha de regressão
    plt.plot(x_fit, y_fit, color='red', label='Regressão Polinomial (grau {}).format(len(coeficientes)-1))

    plt.xlabel('DIAS')
    plt.ylabel('PREÇO EM R$')
    plt.title('Regressão Polinomial')
    plt.legend()
    plt.grid(True)
    plt.show()
```

Acima, foi construída a função de plot, que recebe o x, o y e a lista contendo os coeficientes, observe que no x, já foi estendido para 15 dias a frente dos dados disponíveis, para que seja possível observar graficamente o comportamento do polinômio após o fim dos dados que foram fornecidos. Observe também que foi usado o método polyval para que o polinômio fosse montado.

```
grau_do_polinomio = 8 # Escolha o grau do polinômio desejado
coeficientes = regressao_polinomial(x, y, grau_do_polinomio)

# Printar os coeficientes
print(f'Coeficientes do polinômio de grau {grau_do_polinomio}:')
print(coeficientes)

plot_regressao_polinomial(x, y, coeficientes)

# Calcular os resíduos acumulados
y_pred = np.polyval(coeficientes, x)
residuos = y - y_pred
residuos_acumulados = np.cumsum(residuos)

# Plotar o gráfico dos resíduos acumulados
plt.figure(figsize=(10, 5))
plt.plot(range(len(residuos_acumulados)), residuos_acumulados, marker='o', linestyle='-', color='green')
plt.title('Resíduos Acumulados da Regressão Polinomial')
plt.xlabel('Índice')
plt.ylabel('Resíduos Acumulados')
plt.grid(True)
plt.show()

#previsão do preço
dias_a_estender = 15
x_estendido = np.arange(np.max(x)+1, np.max(x)+1 + dias_a_estender).reshape(-1, 1)
y_estendido = np.polyval(coeficientes, x_estendido)

# Printar os valores previstos para os próximos 15 dias
print(f'Previsão de preço da ação para os próximos {dias_a_estender} dias:')
for i in range(dias_a_estender):
    print(f'Dia {i+1}: {y_estendido[i]}')
```

Por fim, chegamos ao ponto em que são utilizadas as funções construídas. Para o cálculo do resíduo, foi utilizado o método do numpy cumsum, que faz o somatório.

## CONCLUSÃO

Por fim, após todas as análises feitas, é possível que o modelo mais confiável para se seguir é o modelo com o polinômio de grau 6, pois ele não apresenta o **overfitting**. Lembrando que o trabalho é apenas uma exemplificação de uso do método de regressão polinomial por mínimos quadrados, e que basear escolhas financeiras nesses modelos é arriscado.



Veja só, enquanto a previsão encontrada foi de um preço por ação de R\$34,16, na realidade o preço foi de R\$37,21, um valor próximo, demonstrando que a qualidade da previsão e do método aqui desenvolvido foi satisfatória, porém, mesmo assim, é importante salientar o risco embutido em especular se baseando em modelos computacionais.

O risco dessas especulações são altos, mais altos ainda quando tratamos ações como a da PETROBRAS, que são voláteis para os padrões da bolsa brasileira, por conta da influência que sofre do Governo Brasileiro, portanto, a maior dificuldade é justamente essa, é impossível quantificar o erro associado às ações, principalmente brasileiras, por conta de fatores fiscais, políticos e diversos outros, que não serão citados aqui pois não é o objetivo deste documento.



Por fim, utilizando a previsão de grau 6, o investidor deveria optar em não comprar o seu lote de ações PETR4, ou então, se ele tiver um apetite maior ao risco, ele pode operar em short (vendido), para que ganhasse com a desvalorização dos papéis. Ou então, o investidor poderia optar em comprar os papéis mais baratos para aproveitar os dividendos oferecidos pela PETROBRAS, que historicamente, são cavalares. Sendo, nos últimos 12 meses, um yield de 17,67%.