

**Centro Federal de Educação Tecnológica de Minas Gerais - CEFET-MG**

**Disciplina:** Princípio de Modelagem Matemática

**Dia/Horário:** Segunda e Quarta-feira, das 14h50min às 16h30min



# **Análise de Flutuação da Criptomoeda Bitcoin**

## **RELATÓRIO FINAL**

Michelle Hanne Soares de Andrade

Belo Horizonte  
Julho, 2023.

## SUMÁRIO

1- INTRODUÇÃO.....	3
1.1 – Hipótese e Objetivo .....	3
2- REVISÃO BIBLIOGRÁFICA.....	4
3- BASE DE DADOS E SÉRIE TEMPORAL BTC .....	5
3.1 Base de Dados .....	5
3.2 Série Temporal BTC .....	6
4- ALGORITMO PROPOSTO .....	8
4- DISCUSSÃO.....	14
5- CONCLUSÃO .....	22
6- REFERÊNCIAS .....	22

## 1- INTRODUÇÃO

Lançado em 2009, o Bitcoin (BTC ou XBT) é a primeira criptomoeda cujas transações são verificadas e registradas publicamente em um *ledger* distribuído, denominado blockchain. Os blocos de transações são criptografados em um conjunto de funções *hash* SHA-256, formando um registro imutável de todas as transações que já ocorreram (KELSEY AND SCHNEIER, 2005).

Em 2023 a quantidade de emissões de BTC ultrapassa 19 milhões, atingindo 92,47% do total de aproximadamente 21 milhões. A cada quatro anos ocorre o denominado *halving*, que é o corte de 50% na produção de novas moedas do Bitcoin. O *halving* foi projetado para reduzir o efeito inflacionário na emissão<sup>1</sup>.

A oscilação de alta e baixa do BTC atingiu o marco histórico de US\$ 20 mil (1 unidade de Bitcoin) em dezembro de 2017, crescendo absurdamente 1.950%. Porém, ao longo de 2018 teve quedas abruptas de até 83%. Em 2021 atingiu cerca de US\$ 66 mil.

Em 2017, teve início as primeiras negociações de derivativos de BTC pela *Cboe Global Markets*, e posteriormente pela CME Group nos EUA. Os derivativos são instrumentos (contratos) financeiros entre duas partes que concordam com a compra ou venda de um ativo ou mercadoria, definindo preço, quantidade e uma data futura para liquidação financeira e entrega. Praticamente todos os mercados com liquidez relevante (ex. moedas, ações, commodities, etc.) contam com oferta de derivativos, e mais recentemente o BTC.

Por este motivo, compreender a flutuação da série histórica do BTC é essencial para entender o mercado de derivativos, gerar tendências futuras e previsões.

### 1.1 – Hipótese e Objetivo

A hipótese inicial da pesquisa é a de que valores passados do BTC impactam o preço futuro, em curto prazo (dia subsequente) e longo prazo (30 dias, por exemplo).

Com vistas a compreender esta hipótese, têm-se o objetivo geral:

---

<sup>1</sup> <https://buybitcoinworldwide.com/how-many-bitcoins-are-there>

## PRINCÍPIOS DE MODELAGEM MATEMÁTICA

Analisar o coeficiente de Hurst através do método da variância agregada, e assim determinar a memória longa e similaridade na série do BTC.

## 2- REVISÃO BIBLIOGRÁFICA

As análises de séries temporais é um tema recorrente na economia e computação, o que também se aplica ao estudo da série histórica da criptomoeda Bitcoin.

Wu et al (2020) usou o coeficiente de variação para medir a extensão da descentralização para sistemas Bitcoin e Ethereum. Outros autores, como Velankar et al (2018), realizaram a previsão do sinal de mudança de preço diário com a maior precisão possível. Usando regressão bayesiana e modelo linear generalizado, alcançando uma precisão de 51% com o modelo linear generalizado.

A técnica ARIMA (*Auto-Regressive Integrated Moving Averages* - Médias Móveis Integradas e Auto-Regressivas), é comumente usada como estratégia em alguns trabalhos, como McNally et al (2018), Guo and Fantulin (2018) e Mallqui and Fernandes (2018).

Uso de Rede Neural Recorrente (*Recurrent Neural Network* - RNN) e rede de memória de curto prazo (*Long Short Term Memory* - LSTM) são bastante usadas para realizar previsão de séries temporais (MCNALLY, 2016), (MCNALLY et al. 2018), (MALLQUI and FERNANDES, 2018) e (JANG and LEE, 2017).

McNally et al (2018) usou a rede LSTM com o método ARIMA e obteve precisão de 52%, e um erro quadrático médio (RMSE) de 8%. Com uso de Deep Learning, e com o modelo LSTM e RNN, Ji et al (2019) obtiveram precisão de 60%.

## 3- BASE DE DADOS E SÉRIE TEMPORAL BTC

### 3.1 Base de Dados

O conjunto de dados utilizado neste trabalho refere-se ao histórico do mercado de negociações de BTC em intervalos de 1 minuto de 31/12/2011 até 31/03/2021, disponível no sítio Kaggle (<https://www.kaggle.com/mczielinski/bitcoin-historical-data>).

Os atributos existentes no conjunto de dados são: **Timestamp**, **Open**, **High**, **Low**, **Close**, **Volume (BTC)**, **Volume (Currency)** e **Weighted Price**. Alguns registros de minutos não possuem alteração, e estão caracterizados com a sigla NaN. O atributo **Timestamp** é do tipo inteiro e representa a data no formato Unix, os demais atributos são do tipo decimal. O atributo **Open** refere-se ao valor de abertura do BTC, o atributo **Close** ao valor de fechamento. O atributo **High** ao maior valor no minuto e **Low** ao menor valor. O **Volume BTC** refere-se ao volume de negociações da moeda BTC no intervalo de minuto, já o **Volume (Currency)** representa o valor negociado em um minuto ( $\text{Volume BTC} * \text{Weighted Price}$ ). E, por fim, o **Weighted Price** refere-se ao preço ponderado do BTC, calculado para as últimas 24 horas, 7 dias ou 30 dias.

Os dados entre os anos de 2021 e 2023 foram extraídos do sítio Yahoo Finanças<sup>2</sup>. Os dados foram coletados entre 01/04/2021 e 02/07/2023 com os atributos **TimeStamp**, **Open**, **High**, **Low** e **Close**.

Inicialmente foi realizada a limpeza e organização dos dados. Optou-se pelo uso dos atributos **TimeStamp**, **Open**, **High**, **Low** e **Close**, agrupados em 4 tipos de granularidades, conforme Figura 1.

**Figura 1:** Granularidade dos dados

Granularidade	Número de Registros
1 dia	4.202
3 dias	1.401
5 dias	841
10 dias	421

**Fonte:** o autor.


<sup>2</sup> [https://br.financas.yahoo.com/quote/BTC-USD/history/?guccounter=1&guce\\_referrer=aHR0cHM6Ly93d3cuZ29vZ2xLMnVbS8&guce\\_referrer\\_sig=AQAAA FRfw7l8mgr\\_86rbSED64zv3up8BVHLwmXA2KE5hJopSB-Foyy8sBAek7-gGz4tL0znTPFsQMyIaxO6POKxz4ChtlMZt6L-pefDe5fRAKNg4Jx1t6Uta9U3ZWsN-Bdvl6i4R0gnNgCRiMc6rwCp17YQvLY3\\_1mbLjKvHCfoauD1](https://br.financas.yahoo.com/quote/BTC-USD/history/?guccounter=1&guce_referrer=aHR0cHM6Ly93d3cuZ29vZ2xLMnVbS8&guce_referrer_sig=AQAAA FRfw7l8mgr_86rbSED64zv3up8BVHLwmXA2KE5hJopSB-Foyy8sBAek7-gGz4tL0znTPFsQMyIaxO6POKxz4ChtlMZt6L-pefDe5fRAKNg4Jx1t6Uta9U3ZWsN-Bdvl6i4R0gnNgCRiMc6rwCp17YQvLY3_1mbLjKvHCfoauD1)

## PRINCÍPIOS DE MODELAGEM MATEMÁTICA

Durante os experimentos utilizou-se a linguagem de programação Python para o desenvolvimento dos algoritmos, com o uso de bibliotecas como ***stasmmodels***, ***sklearn***, ***numpy***, ***pandas***, ***matplotlib***, entre outras. A Figura 2 exibe uma amostra dos dados.

Foram realizados testes estatísticos para o entendimento da flutuação dos dados, como Teste-F e Dickey-Fuller. Para os testes de Regressão Linear foi utilizada a métrica RMSE (Root Mean Squared Error).

**Figura 2:** Exemplo de uma amostra de dados



```
data.head(10)
```

	Datetime	Open	High	Low	Close
0	2011-12-31	4.39	4.58	4.39	4.58
1	2012-01-01	4.58	5.00	4.58	5.00
2	2012-01-02	5.00	5.00	5.00	5.00
3	2012-01-03	5.32	5.32	5.14	5.29
4	2012-01-04	4.93	5.57	4.93	5.57
5	2012-01-05	5.72	6.65	5.72	6.65
6	2012-01-06	6.65	6.90	6.00	6.00
7	2012-01-07	6.80	6.80	6.80	6.80
8	2012-01-08	6.90	7.00	6.90	7.00
9	2012-01-09	6.99	6.99	6.23	6.30

Fonte: o autor.

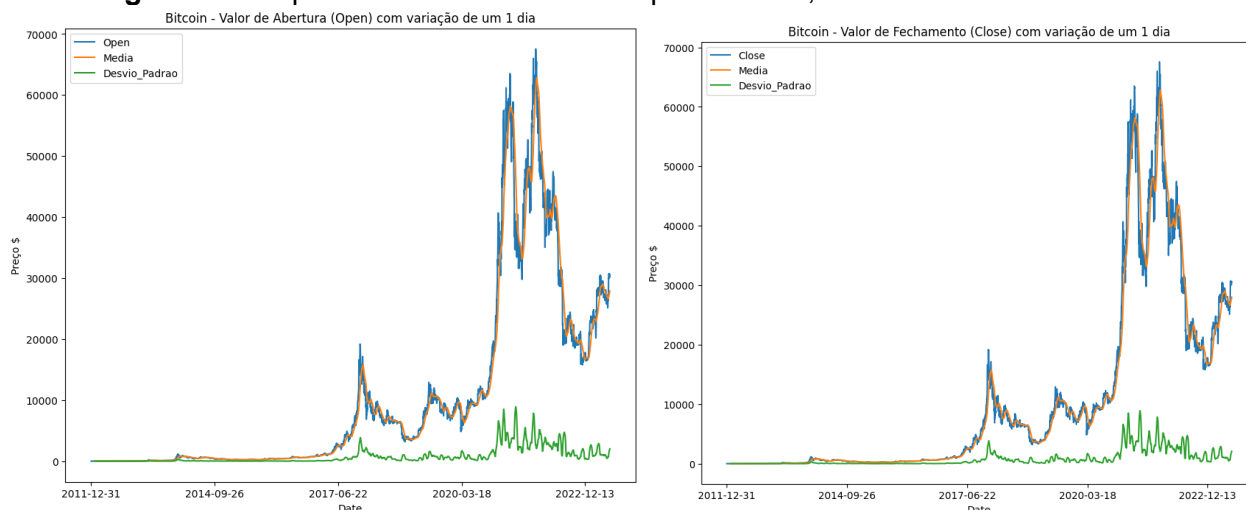
### 3.2 Série Temporal BTC

Os dados históricos do BTC são modelados como uma série temporal. Segundo Montgomery et al (2015), o termo série temporal é definido como uma sequência cronológica de observações sobre um objeto ou evento de interesse.

Uma característica importante deste tipo de dados é que as observações vizinhas são dependentes, e o interesse é analisar e modelar esta dependência. Buscando realizar uma análise sobre a série temporal BTC, fez-se o Teste-F de duas amostras para variâncias de atributos. Analisou-se os valores de *Open* e *Close* (valor de abertura e de fechamento) da série histórica do BTC, com a seguinte hipótese nula, **Ho: as variáveis são estatisticamente iguais**. A Figura 3 mostra o comportamento das variáveis *Close* e *Open*, respectivamente.

## PRINCÍPIOS DE MODELAGEM MATEMÁTICA

**Figura 3:** Comportamento das variáveis Open e Close, incluindo Média e Desvio Padrão



**Fonte:** o autor.

Percebe-se nitidamente semelhanças entre o comportamento das duas séries, assim como os valores de média e desvio padrão. Resultado do Teste-F com nível de significância de 0,05 (5%).

**Estatística F:** 0.0003593445136589722

**Valor-p:** 0.9848763493696489

O resultado da estatística do F mostra que não há evidências para rejeitar a hipótese nula. Assim, as variáveis são estatisticamente iguais.

Com a constatação de que os valores de *Open* e *Close* são estatisticamente iguais, adotou-se a variável **Close** para prosseguir com os testes.

Analisando a Figura 3, percebe-se que a média acompanha a tendência da série temporal (não constante), e o desvio padrão sofre oscilações, ou melhor, não é constante. Essas características mostram que a Série Temporal BTC não é estacionária. Em alguns modelos preditivos para séries temporais, essa é uma característica necessária para a aplicação das técnicas de regressão. Para que uma série seja considerada estacionária, deve-se garantir propriedades estatísticas constantes ao longo do tempo, como: (i) uma média constante, (ii) uma variância constante e (iii) uma autocovariância que não dependa do tempo.

Para avaliar a questão da Série Temporal BTC ser estacionária ou não, realizou-se o Teste de Dickey-Fuller Aumentado (ADF). Neste teste, a presença de uma ou mais raízes unitárias indica não ser estacionária no comportamento de uma série histórica, ou seja, os valores tendem a aumentar com o transcorrer do tempo, assumindo diferentes padrões. Assim, a hipótese nula é

## PRINCÍPIOS DE MODELAGEM MATEMÁTICA

**H0: a Série não é estacionária.** Para se rejeitar H0, a estatística do teste Dickey-Fuller deve ser menor que a estatística dos valores críticos, correspondendo a 1%, 2% e 5%.

No resultado fornecido, a estatística ADF de -1.5856374001835662 é maior que os valores críticos de -3.4319191438819407 (1%), -2.8622333615468443 (5%) e -2.567139082403142 (10%). Isso significa que a estatística ADF não está abaixo desses valores críticos, indicando que não há evidências estatísticas suficientes para rejeitar a hipótese nula. Por tanto, a Série BTC não é estacionária.

```
Resultado do Teste de Dickey-Fuller Aumentado (ADF):  
Estatística ADF: -1.5856374001835662  
Valor-p: 0.49080419976937306  
Valores críticos:  
1%: -3.4319191438819407  
5%: -2.8622333615468443  
10%: -2.567139082403142  
Não há evidências para rejeitar a hipótese nula. A série não é estacionária.
```

**H0: a Série não é Estacionária.**

Algumas estratégias são usadas para transformar a Série em Estacionária, como: (i) Agregação – usando a média para um período de tempo como médias semanais ou mensais, (ii) Suavização – usando médias móveis e (iii) Ajuste polinomial – encaixando um modelo de regressão. Outras técnicas para remover a tendência e sazonalidade podem auxiliar neste processo, como: (i) Diferenciação – considerando a diferença com um determinado intervalo de tempo e (ii) Decomposição – modelando tanto a tendência quanto a sazonalidade e removendo-as a partir do modelo.

Foram avaliadas duas técnicas para ajuste da Série em Estacionária, são elas: diferenciação e suavização, conforme tópicos a seguir.

## 4- ALGORITMO PROPOSTO

O algoritmo proposto segue alguns passos, conforme descrito abaixo:

**Passo 1** – Realiza a regressão linear na variável Close, ajusta os dados para log x log, mostra os resultados e o gráfico do ajuste ao final.

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



## PRINCÍPIOS DE MODELAGEM MATEMÁTICA

```
from scipy.stats import linregress
from scipy.stats import kurtosis, powerlaw, norm
from numpy import cumsum, log, polyfit, sqrt, std, subtract
from numpy.random import randn

# Import data
data = pd.read_csv("/content/drive/MyDrive/Colab Notebooks/Pratica_Fi-
nal_MMC/ohlcl1d.csv")

# Calcular a variação ao longo do tempo (diferença entre os valores de Close)
variation = np.diff(data['Close'])

# Calcular o desvio quadrático médio ao longo do tempo
squared_deviation = variation ** 2
mean_squared_deviation = np.mean(squared_deviation)

# Criar uma lista com os índices do tempo (1, 2, 3, ...)
time = np.arange(1, len(variation) + 1)

# Adicionar uma constante ao vetor de tempo para gerar uma interceptação
constant = np.ones_like(time)

# Filtrar os valores de squared_deviation que são maiores que zero
nonzero_indices = squared_deviation > 0
filtered_squared_deviation = squared_deviation[nonzero_indices]
filtered_time = time[nonzero_indices]
filtered_constant = constant[nonzero_indices]

# Criar a matriz de entrada para a regressão linear
time_with_constant = np.column_stack((filtered_time, filtered_constant))

print(filtered_squared_deviation)
# Converter os valores para logxlog
log_filtered_squared_deviation = np.log(filtered_squared_deviation)
log_time_with_constant = np.log(filtered_time)

# Realizar a regressão linear logxlog
slope, intercept, r_value, p_value, std_err = linregress(log_time_with_cons-
tant, log_filtered_squared_deviation)

# Calcular as estatísticas da regressão linear
slope_std_err = std_err * np.sqrt(len(log_time_with_constant)) # Erro padrão
da inclinação (slope)
r_squared = r_value**2 # Coeficiente de determinação ( $R^2$ )

# Calcular o erro quadrático médio R2
predicted = slope * log_time_with_constant + intercept
residuals = log_filtered_squared_deviation - predicted
```

## PRINCÍPIOS DE MODELAGEM MATEMÁTICA

```
mse = np.mean(residuals ** 2)
r2 = 1.0 - mse / np.var(log_filtered_squared_deviation)

# Imprimir as estatísticas da regressão linear e o erro quadrático médio R2
print("Estatísticas da Regressão Linear:")
print(f"Inclinação (slope): {slope}")
print(f"Interceptação: {intercept}")
print(f"Erro padrão da inclinação: {slope_std_err}")
print(f"Coeficiente de determinação (R²): {r_squared}")

print("\nErro Quadrático Médio R2:")
print(f"Erro Quadrático Médio R2: {r2}")

# Plotar o gráfico de regressão linear logxlog
plt.scatter(log_time_with_constant, log_filtered_squared_deviation, label='Dados')
plt.plot(log_time_with_constant, slope * log_time_with_constant + intercept,
         'r', label='Regressão Linear')
plt.xlabel('Tempo (log)')
plt.ylabel('Desvio Quadrático (log)')
plt.title('Desvio Quadrático Médio ao Longo do Tempo')
plt.legend()
plt.show()
```

**Passo 2 –** Calcula a curtose e ajusta o expoente da Lei de potência ajustado pela curtose, ao final mostra as estatísticas.

```
# Calcular a curtose da variável "Close"
close_data = np.diff(data['Close'])
#close_data = data['Close']
curtose = kurtosis(close_data)

# Calcular o expoente da lei de potência ajustado pela curtose
slope = 1 - (curtose / 6)

# Teste de hipóteses para o expoente da lei de potência com ajuste pela curtose
null_hypothesis = "Random Walk é verdadeiro (slope = 1)"
if slope == 1:
    result = "Hipótese nula confirmada"
else:
    result = "Hipótese nula rejeitada"

# Plotar histograma das posições finais das caminhadas (Close)
plt.figure(figsize=(10, 6))
plt.hist(close_data, bins=30, density=True, alpha=0.5, label='Posições Finais')
plt.xlabel('Close')
plt.ylabel('Densidade')

# Estimar densidade pela distribuição normal
```

## PRINCÍPIOS DE MODELAGEM MATEMÁTICA

```
mu, std = norm.fit(close_data)
x = np.linspace(min(close_data), max(close_data), 100)
y = norm.pdf(x, mu, std)
plt.plot(x, y, 'r-', label='Distribuição Normal')

# Configurar título e legenda do gráfico
plt.title('Histograma das Posições Finais com Densidade pela Distribuição Normal')
plt.legend()

# Exibir o gráfico
plt.show()

# Imprimir a curtose, o expoente da lei de potência ajustado e o resultado do teste de hipóteses
print('Curtose:', curtose)
print('Expoente da Lei de Potência Ajustado:', slope)
print('Hipótese Nula:', null_hypothesis)
print('Resultado do Teste de Hipóteses:', result)
```

**Passo 3 – Comparar a série temporal do BTC com as caminhadas aleatórias do Random Walk.**

Mostrar o gráfico ao final.

```
# Calcular a variação ao longo do tempo (diferença entre os valores de Close)
variation = np.diff(data['Close'])

# Gerar caminhadas aleatórias (random walk)
num_walks = 5 # Número de caminhadas aleatórias a serem geradas
num_steps = len(variation) # Número de passos na caminhada (mesmo que a variação)
random_walks = np.cumsum(np.random.choice([-1, 1], size=(num_steps, num_walks)), axis=0)

# Plotar um gráfico comparando a variação do bitcoin com as caminhadas aleatórias
plt.figure(figsize=(12, 6))
plt.plot(variation, label='Variação do Bitcoin', linewidth=2)
plt.plot(random_walks, alpha=0.7)
plt.xlabel('Tempo')
plt.ylabel('Variação')
plt.title('Variação do Bitcoin vs. Caminhadas Aleatórias')
plt.legend(['Variação do Bitcoin'] + [f'Caminhada Aleatória {i+1}' for i in range(num_walks)])
plt.grid(True)
plt.show()
```

## PRINCÍPIOS DE MODELAGEM MATEMÁTICA

**Passo 4** – Aplicação de duas técnicas – **Diferenciação e Suavização** com média móveis, no exemplo abaixo tem-se 7 dias.

```
# Realizar a diferenciação na coluna 'Close'
data['Close_diff'] = data['Close'].diff()

# Suavização - médias móveis com janela de 7 dias
data['Close_smooth'] = data['Close_diff'].rolling(window=7).mean()

data['Close']=data['Close_smooth']
# Remover a primeira linha que terá um valor NaN após a diferenciação
data = data.dropna()
# Salvar os dados transformados em um novo arquivo CSV
data.to_csv('/content/drive/MyDrive/Colab Notebooks/Pratica_Final_MMC/ohlclld_variation.csv', index=False)

# Plotar a variação do preço Close ao longo do tempo
plt.figure(figsize=(12, 6))
plt.plot(data['Datetime'], data['Close_smooth'])
plt.xlabel('Data')
plt.ylabel('Variação do Preço Close')
plt.title('Variação do Preço Close do Bitcoin ao Longo do Tempo')
plt.grid(True)

# Exibir o gráfico
plt.show()
```

**Passo 5** – O arquivo gerado nas técnicas de diferenciação e suavização foi utilizado para realizar o teste de Dickey-Fuller, validando se a série é estacionária.

```
data = pd.read_csv("/content/drive/MyDrive/Colab Notebooks/Pratica_Final_MMC/ohlclld_variation.csv")
from statsmodels.tsa.stattools import adfuller
# Selecionar a coluna 'Close'
close_data = data['Close_diff']
# Executar o teste de Dickey-Fuller Aumentado (ADF)
result = adfuller(close_data)

# Extrair as estatísticas do teste
adf_statistic = result[0]
p_value = result[1]
critical_values = result[4]

# Imprimir o resultado do teste
```

## PRINCÍPIOS DE MODELAGEM MATEMÁTICA

```
alpha = 0.05 # Nível de significância

print("Resultado do Teste de Dickey-Fuller Aumentado (ADF):")
print("Estatística ADF:", adf_statistic)
print("Valor-p:", p_value)
print("Valores críticos:")
for key, value in critical_values.items():
    print(f"{key}: {value}")

if adf_statistic < critical_values['5%']:
    print("Rejeitar a hipótese nula. A série é estacionária.")
else:
    print("Não há evidências para rejeitar a hipótese nula. A série não é esta-
cionária.")
```

**Passo 6** – Executa o teste do coeficiente de Hurst usando o método da variância agregada em janelas deslizantes de tamanho 100. Ao final mostra o resultado do coeficiente e o gráfico.

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

# Calcular a diferença entre os valores de fechamento (close)
close_diff = data['Close'].diff().dropna()

# Função para calcular o coeficiente de Hurst usando o método da Variância
Agregada
def hurst_coefficient(close_diff, window_size):
    total_size = len(close_diff)
    subseries = [close_diff[i:i+window_size] for i in range(0, total_size, win-
dow_size)]
    avg_variances = []

    for subset in subseries:
        subset_size = len(subset)
        subset_mean = sum(subset) / subset_size
        deviations = [val - subset_mean for val in subset]
        cumulative_sum = [sum(deviations[:i+1]) for i in range(subset_size)]
        cumulative_range = max(cumulative_sum) - min(cumulative_sum)
        subset_variance = sum([dev**2 for dev in deviations]) / subset_size
        avg_variance = subset_variance / cumulative_range
        avg_variances.append(avg_variance)

    return sum(avg_variances) / len(avg_variances)

# Calcular o coeficiente de Hurst com janelas deslizantes de tamanho 100
window_size = 100
coefficients = []
```

## PRINCÍPIOS DE MODELAGEM MATEMÁTICA

```
for i in range(len(close_diff) - window_size + 1):
    subset = close_diff[i:i+window_size]
    coefficient = hurst_coefficient(subset, window_size)
    coefficients.append(coefficient)
    # print(f"Janela Deslizante {i+1}: Coeficiente de Hurst = {coefficient}")

# Plotar o gráfico da variação do coeficiente de Hurst
plt.plot(coefficients)
plt.xlabel('Janela Deslizante')
plt.ylabel('Coeficiente de Hurst')
plt.title('Variação do Coeficiente de Hurst')
plt.show()

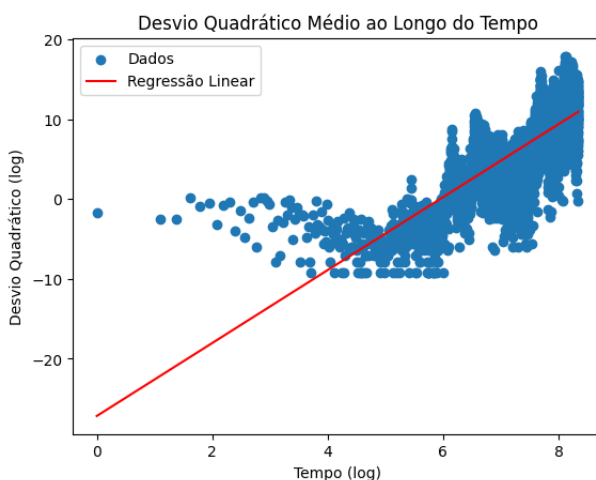
# Calcular o coeficiente de Hurst para toda a série
hurst = hurst_coefficient(close_diff, 100)
print(f"Coeficiente de Hurst para a série temporal: {hurst}")
```

## 4- DISCUSSÃO

O resultado abaixo mostra o Desvio Quadrático Médio ao longo do Tempo do período 31/12/2011 até 02/07/2023. Os testes de Regressão linear e Curtose foram realizados nos períodos de agrupamentos diferentes de 1, 3, 5 e 10 dias.

Percebe-se que o Coeficiente de determinação diminui e o erro quadrático médio aumenta.

### Agrupamento de 1 dia dos valores originais da variável Close da série BTC



#### Estatísticas da Regressão Linear 1 DIA:

Inclinação (slope): 4.565997210514398

Interceptação: -27.163011046574294

Erro padrão da inclinação:

3.687222054487618

Coeficiente de determinação ( $R^2$ ):

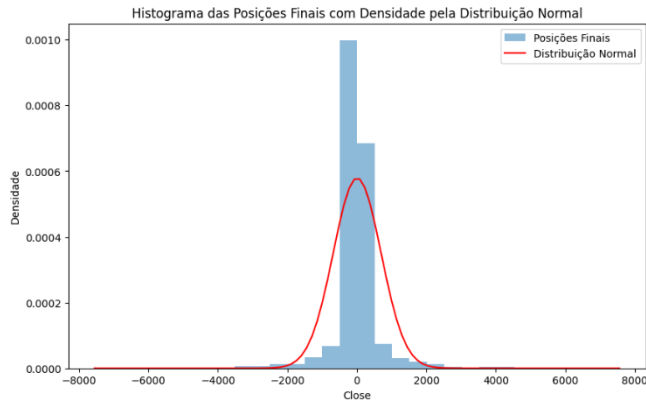
0.6053975287526773

Erro Quadrático Médio R2:

Erro Quadrático Médio R2:

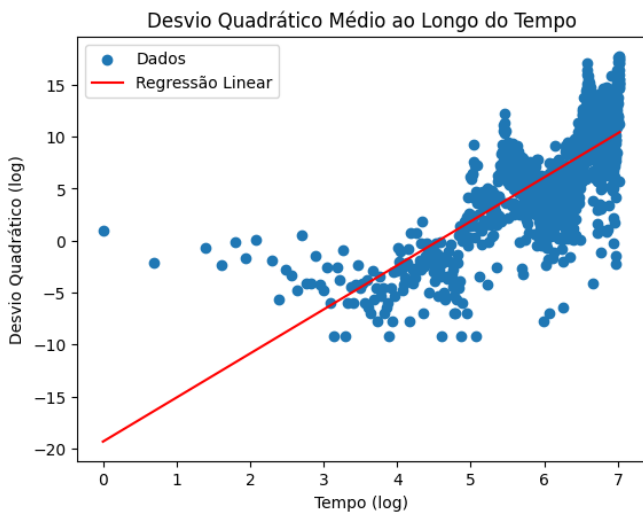
0.6053975287526769

# PRINCÍPIOS DE MODELAGEM MATEMÁTICA



Curtose: 23.160027396040988  
Expoente da Lei de Potência Ajustado: -2.860004566006831  
Hipótese Nula: Random Walk é verdadeiro (slope = 1)  
Resultado do Teste de Hipóteses:  
Hipótese nula rejeitada

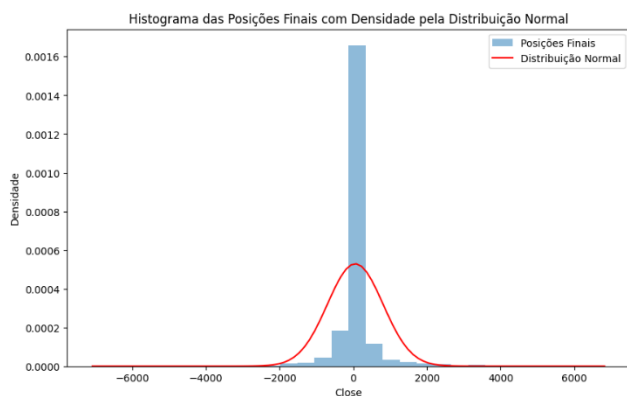
## Agrupamento de 3 dias dos valores originais da variável Close da série BTC



### Estatísticas da Regressão Linear 3 DIAS:

Inclinação (slope): 4.236843472121268  
Interceptação: -19.347576249614658  
Erro padrão da inclinação: 3.768875992125309  
Coeficiente de determinação ( $R^2$ ): 0.5586946309909417

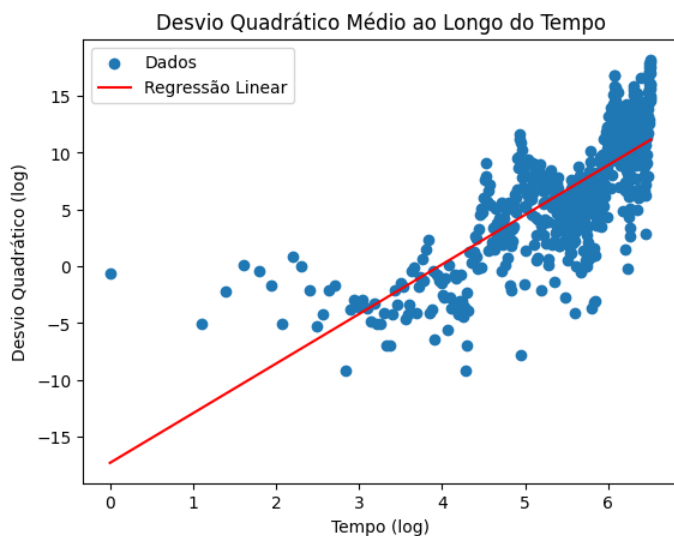
Erro Quadrático Médio R2:  
Erro Quadrático Médio R2:  
0.5586946309909407



Curtose: 34.96116334231004  
Expoente da Lei de Potência Ajustado: -4.826860557051673  
Hipótese Nula: Random Walk é verdadeiro (slope = 1)  
Resultado do Teste de Hipóteses:  
Hipótese nula rejeitada

# PRINCÍPIOS DE MODELAGEM MATEMÁTICA

## Agrupamento de 5 dias dos valores originais da variável Close da série BTC



### Estatísticas da Regressão Linear 5

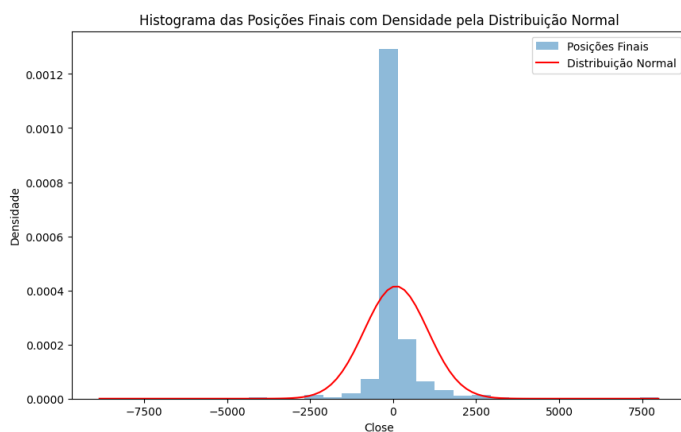
#### DIAS:

Inclinação (slope): 4.368929272010277  
Interceptação: -17.31445567692156  
Erro padrão da inclinação: 3.7029579870614517  
Coeficiente de determinação ( $R^2$ ): 0.5826732521598366

Erro Quadrático Médio R2:

Erro Quadrático Médio R2:

0.5826732521598366



Curtose: 32.80509722979125

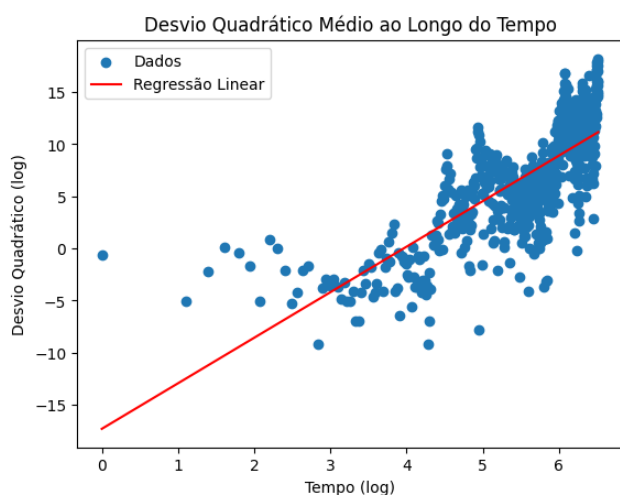
Expoente da Lei de Potência Ajustado: -4.467516204965208

Hipótese Nula: Random Walk é verdadeiro (slope = 1)

Resultado do Teste de Hipóteses:

Hipótese nula rejeitada

## Agrupamento de 10 dias dos valores originais da variável Close da série BTC



### Estatísticas da Regressão Linear 10

#### DIAS:

Inclinação (slope): 4.452294612946602  
Interceptação: -13.609784880319047  
Erro padrão da inclinação: 3.456564721591105  
Coeficiente de determinação ( $R^2$ ): 0.6253358635589604

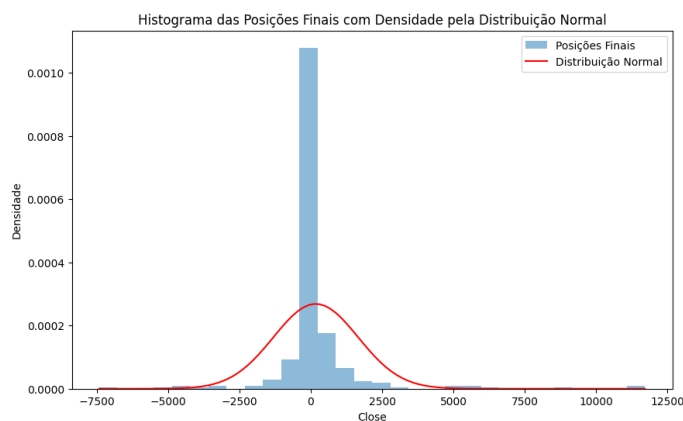
Erro Quadrático Médio R2:

Erro Quadrático Médio R2:

0.6253358635589603



## PRINCÍPIOS DE MODELAGEM MATEMÁTICA



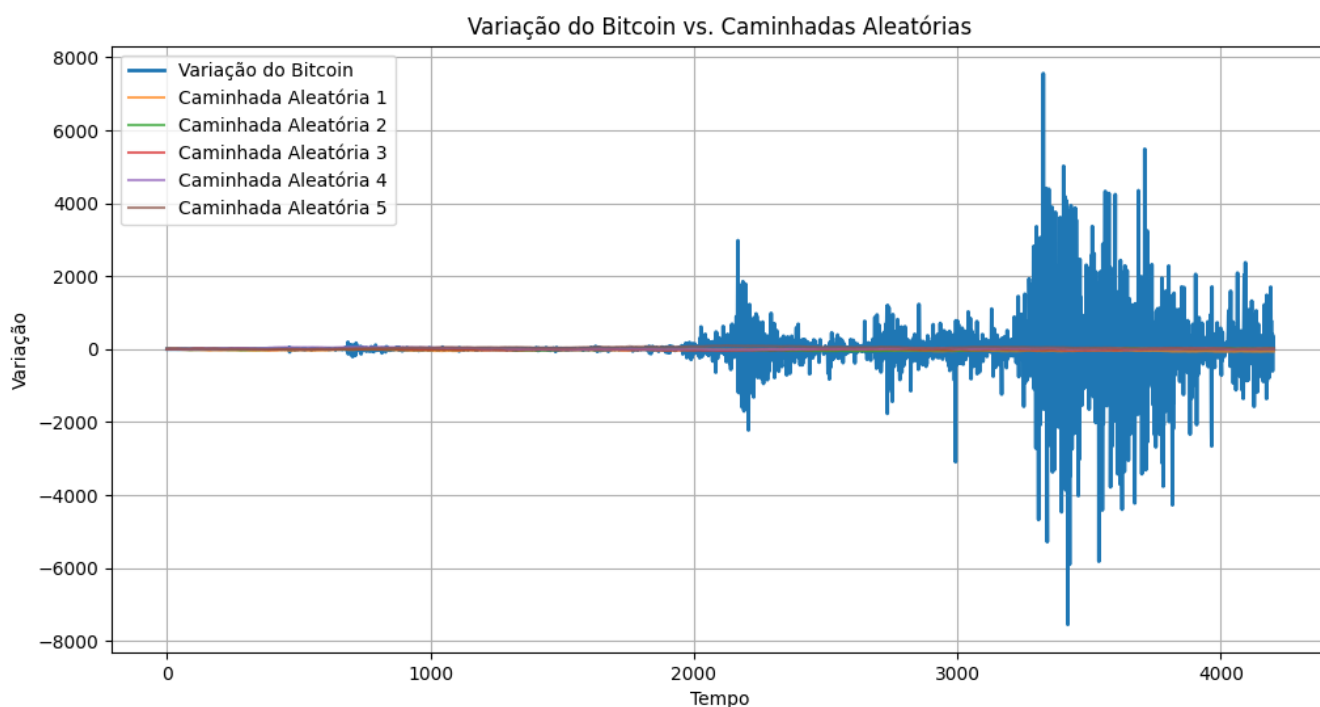
Curtose: 25.79407820782122

Expoente da Lei de Potência Ajustado: -3.29901303463687

Hipótese Nula: Random Walk é verdadeiro (slope = 1)

Resultado do Teste de Hipóteses:  
Hipótese nula rejeitada

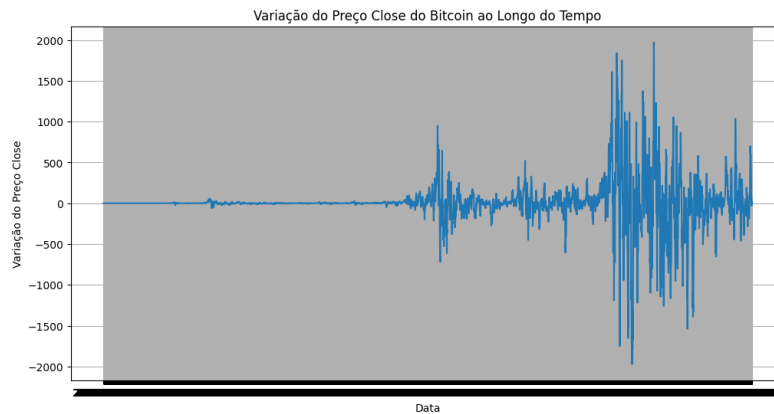
Para prosseguir com a execução dos demais testes optou-se por usar o conjunto de dados agrupado em 1 dia. O gráfico abaixo mostra a comparação do Bitcoin com as caminhadas aleatórias do Random Walk. No Random Walk, as caminhadas tendem a se mover ao longo de uma linha reta, próxima ao eixo zero. Assim, o Random Walk é um processo estocástico, em que cada ponto é a soma do ponto anterior e de um ruído aleatório. Enquanto que a série do BTC não possui comportamento similar ao do Random Walk, comprovando que a série não é estacionária.



Para transformar a série em estacionária foi aplicado dois métodos: diferenciação e suavização. Abaixo a comparação da suavização em médias móveis com períodos diferentes.

# PRINCÍPIOS DE MODELAGEM MATEMÁTICA

## Diferenciação e Suavização Médias móveis de 7 dias



Resultado do Teste de Dickey-Fuller Aumentado (ADF):

Estatística ADF: -10.427980652273147

Valor-p: 1.638664855420527e-18

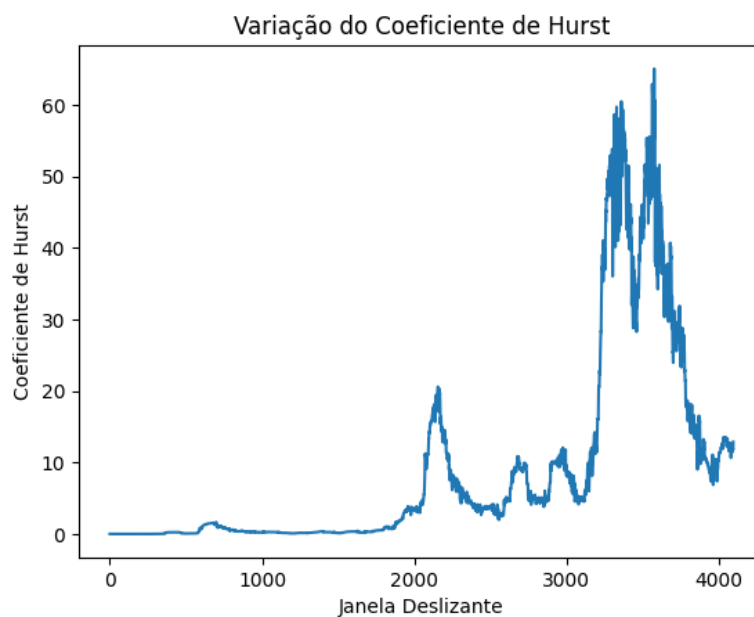
Valores críticos:

1%: -3.4319217839951306

5%: -2.862234527832183

10%: -2.567139703279794

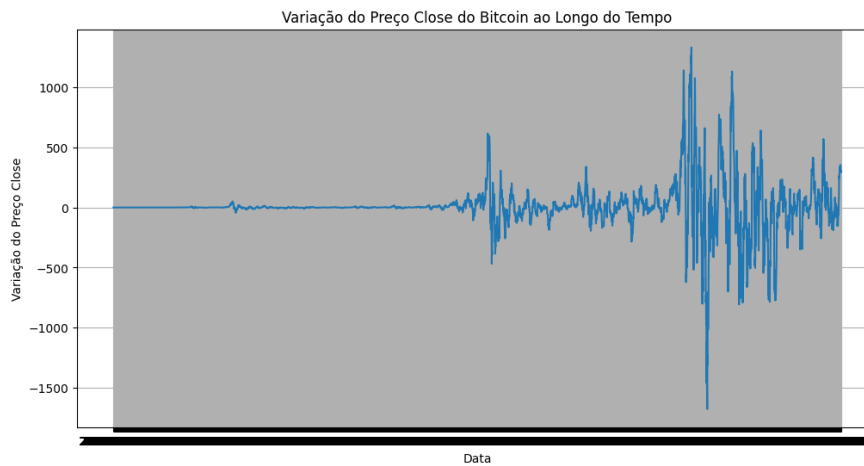
Rejeitar a hipótese nula. A série é estacionária.



Coeficiente de Hurst para a série temporal: 8.783324422136438

# PRINCÍPIOS DE MODELAGEM MATEMÁTICA

## Diferenciação e Suavização Médias móveis de 14 dias



Resultado do Teste de Dickey-Fuller Aumentado (ADF):

Estatística ADF: -10.419170756578877

Valor-p: 1.7225804021202736e-18

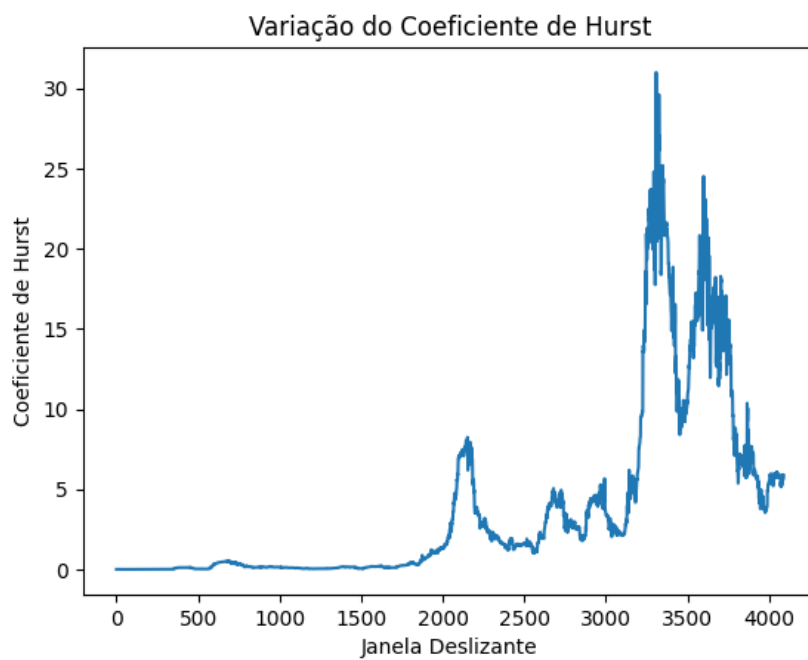
Valores críticos:

1%: -3.4319244330073735

5%: -2.8622356980476913

10%: -2.567140326248864

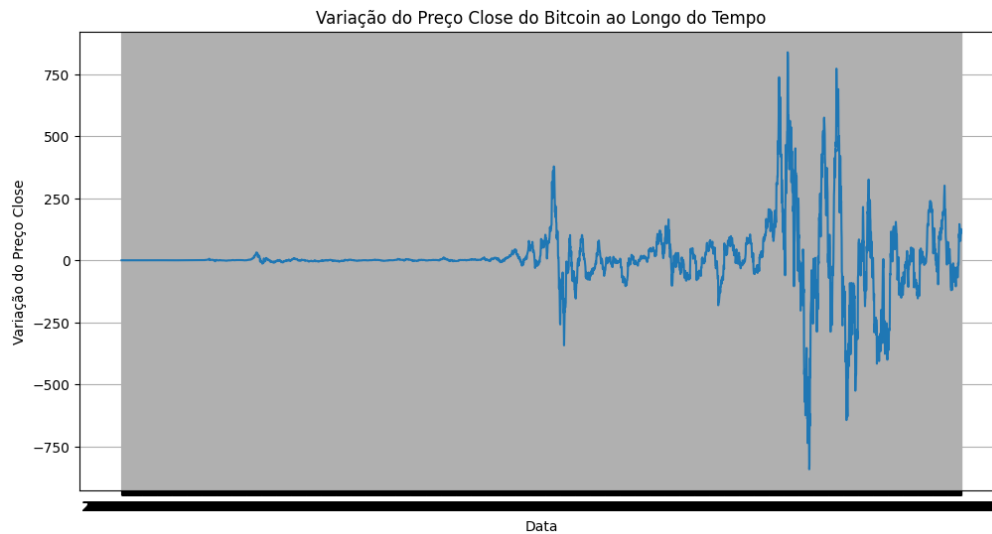
Rejeitar a hipótese nula. A série é estacionária.



Coeficiente de Hurst para a série temporal: 3.613637353831187

# PRINCÍPIOS DE MODELAGEM MATEMÁTICA

## Diferenciação e Suavização Médias móveis de 30 dias



Resultado do Teste de Dickey-Fuller Aumentado (ADF):

Estatística ADF: -10.399003925603465

Valor-p: 1.931356297278316e-18

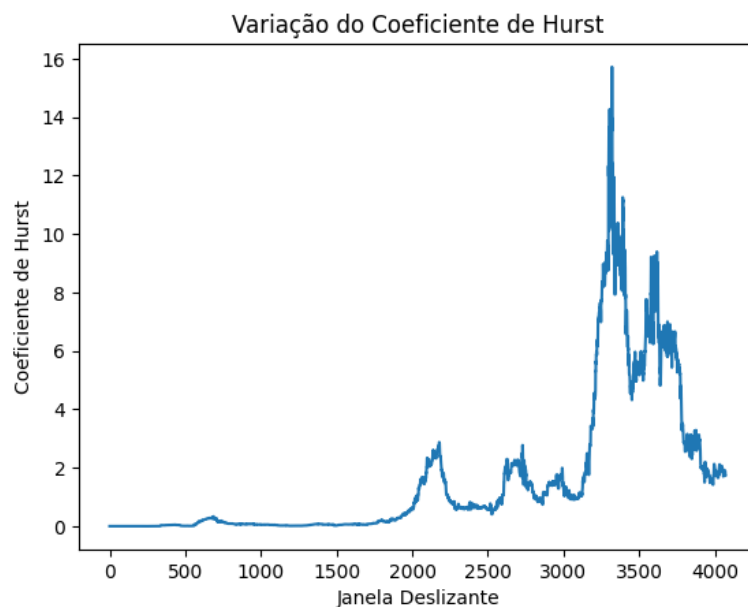
Valores críticos:

1%: -3.431930521551653

5%: -2.8622383876911655

10%: -2.567141758092371

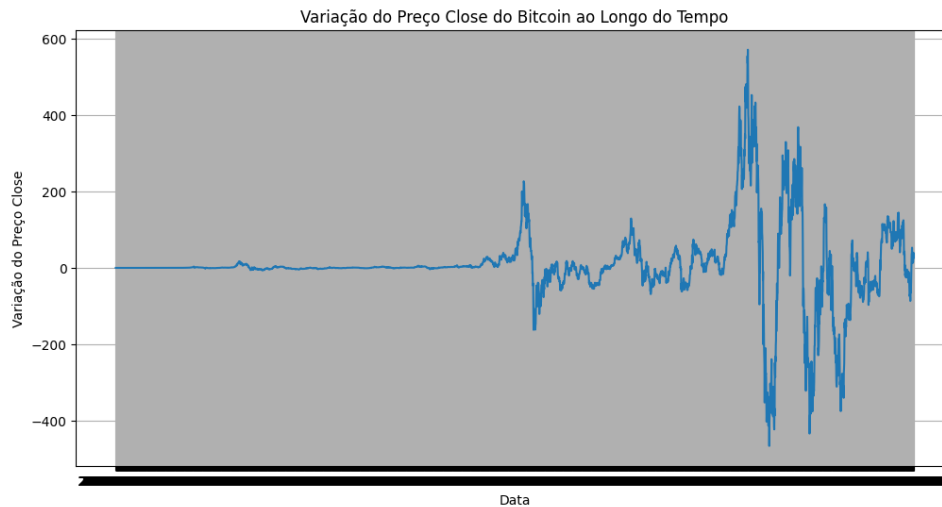
Rejeitar a hipótese nula. A série é estacionária.



Coeficiente de Hurst para a série temporal: 1.7339310894696216

# PRINCÍPIOS DE MODELAGEM MATEMÁTICA

## Diferenciação e Suavização Médias móveis de 60 dias



Resultado do Teste de Dickey-Fuller Aumentado (ADF):

Estatística ADF: -10.361085339277516

Valor-p: 2.395539147548422e-18

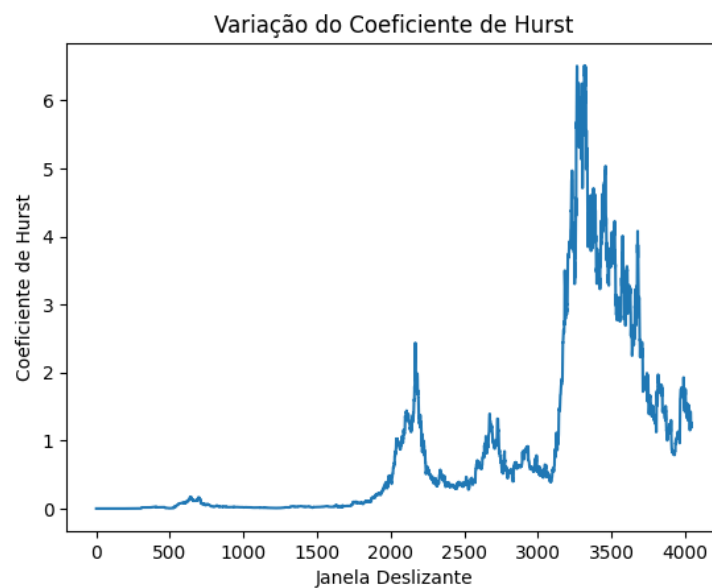
Valores críticos:

1%: -3.431942065422698

5%: -2.8622434872361695

10%: -2.5671444728600945

Rejeitar a hipótese nula. A série é estacionária.



Coeficiente de Hurst para a série temporal: 0.8912227455745423

Nota-se que à medida que aumenta o número de dias das médias móveis, o coeficiente de Hurst tende a diminuir. Analisando os intervalos do coeficiente de Hurst, temos como melhor resultado o valor de 0.89 com uma suavização de 60 dias. Podemos dizer que esta série tende a ter um comportamento de persistência.

## PRINCÍPIOS DE MODELAGEM MATEMÁTICA

- $0 < H < 0,5$ : tem-se uma série anti-persistente, em que um aumento tenderá a ser seguido por um decrescimento, ou vice-versa.
- $H = 0,5$ : a série tem um comportamento aleatório e os valores da série atual não se torna dependente dos valores da série do passado.
- $0,5 < H < 1$ : descreve um comportamento de persistência, em que um decrescimento tenderá a seguir de um decrescimento.

## 5- CONCLUSÃO

Entende-se que é possível estabelecer uma série de BTC estacionária, com um comportamento que tende a ser persistente em agrupamentos superiores a 60 dias. Porém, os valores de BTC podem ser influenciados por diversos fatores, como notícias, eventos econômicos e comportamento dos investidores, o que tende a gerar oscilações bruscas.

Portanto, é importante interpretar os resultados com cautela e considerar outros aspectos ao analisar o comportamento da variação do BTC.

## 6- REFERÊNCIAS

- AMJAD, M. and Shah, D. (2017). Trading bitcoin and online time series prediction. In Anava, O., Khaleghi, A., Cuturi, M., Kuznetsov, V., and Rakhlin, A., editors, Proceedings of the Time Series Workshop at NIPS 2016, volume 55 of Proceedings of Machine Learning Research, pages 1–15, Barcelona, Spain. PMLR.
- C.A. Mallqui, D. and Fernandes, R. (2018). Predicting the direction, maximum, minimum and closing prices of daily bitcoin exchange rate using machine learning techniques. Applied Soft Computing, 75.
- CIAIAN, P., Rajcaniova, M., and Kancs, d. (2016a). The digital agenda of virtual currencies: Can bitcoin become a global currency? Information Systems and e-Business Management, 14.
- Ciaian, P., Rajcaniova, M., and Kancs, d. (2016b). The economics of bitcoin price formation. Applied Economics, 48:1799–1815.
- DUTTA, A.; Kumar, S.; Basu, M. A Gated Recurrent Unit Approach to Bitcoin Price Prediction. J. Risk Financ. Manag. 2020, 13, 23.
- DYHRBERG, A., of Economics, U. S., and University College, D. C. f. E. R. (2015). Hedging Capabilities of Bitcoin: Is it the Virtual Gold? Working paper (University College, Dublin. Centre for Economic Research).

## PRINCÍPIOS DE MODELAGEM MATEMÁTICA

GANDAL, N., Hamrick, J., Moore, T., and Oberman, T. (2017). Price manipulation in the bitcoin ecosystem. Number 12061.

GUO, T. and Antulov-Fantulin, N. (2018). Predicting short-term bitcoin price fluctuations from buy and sell orders.

HOW MANY BITCOINS ARE THERE? Disponível em  
<<https://buybitcoinworldwide.com/howmany-bitcoins-are-there/>>. Acesso em 03 jul 2023.

JANG, H. and Lee, J. (2017). An empirical study on modeling and prediction of bitcoin prices with bayesian neural networks based on blockchain information. IEEE Access, pages 1–1.

JI, S.; Kim, J.; Im, H. A Comparative Study of Bitcoin Price Prediction Using Deep Learning. Mathematics 2019, 7, 898.

KELSEY, J. and Schneier, B. (2005). Second preimages on  $n$ -bit hash functions for much less than  $2^n$  work. In Proceedings of the 24th Annual International Conference on Theory and Applications of Cryptographic Techniques, EUROCRYPT'05, pages 474–490, Berlin, Heidelberg. Springer-Verlag.

KRISTOUFEK, L. (2014). What are the main drivers of the bitcoin price? evidence from wavelet coherence analysis. PLoS ONE.

Li, X. and Alex Wang, C. (2016). The technology and economic determinants of cryptocurrency exchange rates: The case of bitcoin. Decision Support Systems, 95.

MADAN, I., Saluja, S., and Zhao, A. (2014). Automated bitcoin trading via machine learning algorithms. volume 20, pages 1–5.

MCNALLY, S.; Roche, J.; Caton, S. Predicting the Price of Bitcoin Using Machine Learning. In Proceedings of the 26th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing, Cambridge, UK, 21–23 March 2018.

MCNALLY, S. (2016). Predicting the price of bitcoin using machine learning. page 23, Dublin, National College of Ireland. Masters thesis.

Mcnally, S., Roche, J., and Caton, S. (2018). Predicting the price of bitcoin using machine learning. pages 339–343.

MONTGOMERY, D. C., Jennings, C. L., and Kulahci, J. (2015). Introduction to time series analysis and forecasting. New Jersey. John Wiley and Sons.

NAKAMOTO, S. (2009). Bitcoin: A peer-to-peer electronic cash system.

POLASIK, M., Piotrowska, A., Wisniewski, T., Kotkowski, R., and Lightfoot, G. (2014). Price fluctuations and the use of bitcoin: An empirical inquiry. SSRN Electronic Journal.

VELANKAR, S.; Valecha, S.; Maji, S. Bitcoin Price Prediction using Machine Learning. In Proceedings of the 20th International Conference on Advanced Communications Technology (ICACT), Chuncheon-si, Korea, 11–14 February 2018.

WU, Keke et al. A Coefficient of Variation Method to Measure the Extents of Decentralization for Bitcoin and Ethereum Networks. Int. J. Netw. Secur., v. 22, n. 2, p. 191-200, 2020.