

# Aplicação de Simulações em Distribuições

July 23, 2022

## 0.1 Distribuição Normal Com o Criptoativo Bitcoin

O Mercado Bitcoin, maior corretora de criptomoedas da América Latina, disponibiliza uma [API pública](#) onde é possível consultar o resumo diário de negociações realizadas para um dado criptoativo.

É possível coletar os dados históricos de um determinado ativo ao longo do tempo, utilizando-se a seguinte [chamada](#) pretende-se coletar os dados históricos do ativo **Bitcoin** ao longo do mês de junho de 2022. Para cada dia será necessário uma chamada diferente ao endpoint citado.

Assim, assumindo uma **Distribuição Normal** tem-se a seguinte equação:

$$\begin{cases} \mathbb{P}_X(X = x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \\ \mathbb{P}_X = \mu \\ \mathbb{V}_X = \sigma^2 \end{cases} \quad (1)$$

Passemos primeiramente a coleta de dados sobre o ativo Bitcoin

```
[8]: import requests
import pandas as pd
```

```
[12]: month_summary_bitcoin = []
url_day_summary = "https://www.mercadobitcoin.net/api/{ativo}/day-summary/2022/
↪{mes}/{dia}/"
month = 6
for i in range(1,31):
    url_day_formatted = url_day_summary.format(ativo="BTC", mes=month, dia=i)
    response = requests.get(url=url_day_formatted)
    month_summary_bitcoin.append(response.json())
df = pd.DataFrame(month_summary_bitcoin)
df.head()
```

```
[12]:
```

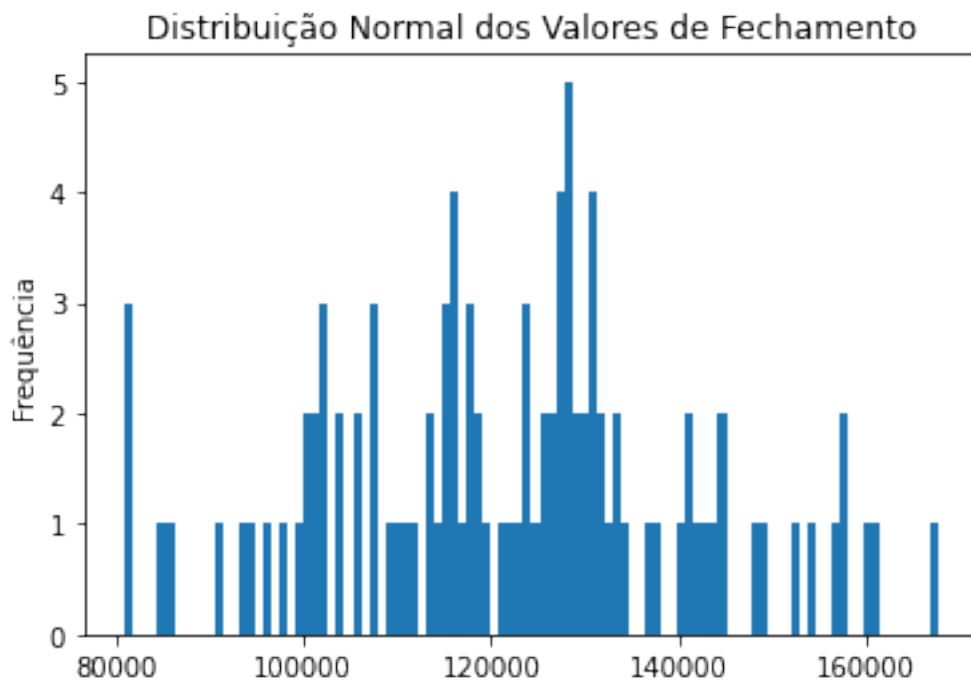
	date	opening	closing	lowest	highest	\
0	2022-06-01	151442.00000	144596.327490	142396.02574	152000.00000	
1	2022-06-02	144612.99999	146833.630260	142032.35493	147753.00000	
2	2022-06-03	146999.99994	142062.761108	141418.60000	147000.00000	
3	2022-06-04	142157.99994	143362.660399	141910.00005	143999.99930	
4	2022-06-05	143357.58202	148500.020000	142227.39202	149108.27999	

	volume	quantity	amount	avg_price
0	6945888.32136877	47.31859469	4621	146789.826851
1	5790208.37420435	39.99302691	3204	144780.448533
2	3871539.36461533	27.09075657	3134	142909.975755
3	998502.68282889	6.99153215	1814	142816.003904
4	1528555.15254143	10.50469379	2163	145511.633475

Agora à análise estatística

```
[14]: import numpy as np
import matplotlib.pyplot as plt
```

```
[15]: x = np.random.normal(df["closing"].mean(), df["closing"].std(), 100)
plt.gca().set(title="Distribuição Normal dos Valores de Fechamento",
              ylabel="Frequência")
plt.hist(x, 100)
plt.show()
```



```
[16]: plt.plot(df.index, df["closing"])
plt.gca().set(title="Valores de Fechamento do Bitcoin", xlabel="Dia",
              ylabel="Valor de Fechamento em R$")
plt.show()
```

