

# ANOVA Em Dados Financeiros do Mercado de Criptoativos

August 7, 2022

## 0.1 ANOVA Em Dados Financeiros do Mercado de Criptoativos

## 0.2 Importando as Bibliotecas

```
[14]: import requests

import numpy as np
import pandas as pd

import scipy.stats as stats
from scipy.stats import levene, bartlett, mannwhitneyu, kruskal

import statsmodels.api as sm
import matplotlib.pyplot as plt
```

### 0.2.1 Construindo uma Base de Dados Financeiros com Ações Do Mercado De Criptoativos

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.

Realizando a seguinte [chamada](#) tem-se os dados de negociação (compra e venda) do criptoativo *Bitcoin* (BTC) para o dia 29/06/2022. Desta maneira, serão realizadas chamadas a API pública com intuito de construir uma Base de Dados com os ativos BITCOIN(BTC), CARDANO (ADA), LITECOIN(LTC), CHILIZ(CHZ), USDC, NANO(LINK) ao longo dos anos de 2020 a 2022, em todos os dias e meses do ano.

```
[2]: ativos = {"BITCOIN": "BTC", "CARDANO": "ADA", "LITECOIN": "LTC", "CHILIZ": "CHZ", "USDC": "USDC", "NANO": "LINK"}
anos = [2020, 2021, 2022]
meses = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
dias = list(range(1, 32))
```

```
[3]: resumo_ativos = []
url_resumo_diario = "https://www.mercadobitcoin.net/api/{ativo}/day-summary/{ano}/{mes}/{dia}/"

for ativo in ativos.values():
```

```

print(ativo)
resumo_ativo = []
for ano in anos:
    print(ano)
    for mes in meses:
        # pular os meses futuros do ano de 2022
        if ano == 2022 and mes > 7:
            continue
        for dia in dias:
            # pular os dias futuros de julho de 2022
            if ano == 2022 and mes == 7 and dia >= 25:
                continue
            url_dia_formatada = url_resumo_diario.
↪format(ativo=ativo,ano=ano, mes=mes, dia=dia)
            resposta = requests.get(url=url_dia_formatada)
            if resposta.status_code == 200:
                resumo_ativo.append(resposta.json())
            else:
                continue
        resumo_ativos.append(resumo_ativo)

```

BTC  
 2020  
 2021  
 2022  
 ADA  
 2020  
 2021  
 2022  
 LTC  
 2020  
 2021  
 2022  
 CHZ  
 2020  
 2021  
 2022  
 USDC  
 2020  
 2021  
 2022  
 LINK  
 2020  
 2021  
 2022

```
[ ]: resumo_ativos
```

```
[5]: acoes_df = pd.DataFrame()

for ativo, resumo in zip(ativos, resumo_ativos):
    indice = [r["date"] for r in resumo]
    acoes_df[ativo] = pd.DataFrame(resumo)["closing"]

# acoes_df["Date"] = [r["date"] for r in resumo_ativos[0]]
acoes_df
```

```
[5]:
```

	BITCOIN	CARDANO	LITECOIN	CHILIZ	USDC	NANO
0	29118.000000	12.89956	167.09015	0.07000	5.44500	60.37949
1	28010.000000	12.43990	160.94876	0.06800	5.38850	62.54979
2	29670.481710	12.40000	170.29813	0.06540	5.46790	64.27989
3	30000.991580	12.33131	173.06007	0.06510	5.44001	65.48000
4	30199.990000	12.07408	177.90000	0.06439	5.39874	65.57001
..	...	...	...	...	...	...
949	125000.000000	NaN	304.00004	NaN	NaN	NaN
950	126708.000000	NaN	318.47012	NaN	NaN	NaN
951	126017.000000	NaN	311.44263	NaN	NaN	NaN
952	124115.769950	NaN	313.65668	NaN	NaN	NaN
953	121148.148952	NaN	311.61799	NaN	NaN	NaN

[954 rows x 6 columns]

```
[6]: for coluna in acoes_df:
    acoes_df[coluna].fillna(acoes_df[coluna].mean(), inplace=True)
acoes_df
```

```
[6]:
```

	BITCOIN	CARDANO	LITECOIN	CHILIZ	USDC	NANO
0	29118.000000	12.899560	167.09015	0.070000	5.445000	60.379490
1	28010.000000	12.439900	160.94876	0.068000	5.388500	62.549790
2	29670.481710	12.400000	170.29813	0.065400	5.467900	64.279890
3	30000.991580	12.331310	173.06007	0.065100	5.440010	65.480000
4	30199.990000	12.074080	177.90000	0.064390	5.398740	65.570010
..	...	...	...	...	...	...
949	125000.000000	5.946881	304.00004	1.091287	5.359013	115.072929
950	126708.000000	5.946881	318.47012	1.091287	5.359013	115.072929
951	126017.000000	5.946881	311.44263	1.091287	5.359013	115.072929
952	124115.769950	5.946881	313.65668	1.091287	5.359013	115.072929
953	121148.148952	5.946881	311.61799	1.091287	5.359013	115.072929

[954 rows x 6 columns]

## 0.2.2 Teste de Homogeneidade da variância do Erro

Pretende-se comparar os criptoativos em relação ao USDC. Por se tratar de uma *stablecoin*, isto é, tem o seu valor lastreado em uma moeda fiduciária, neste caso, o Dólar Americano. A criptomoeda

USDC servirá de base de comparação com as demais, tendo em vista que o seu valor representa o valor do Dólar Americano à época.

```
[7]: levene_df = pd.DataFrame(columns=[k for k in ativos.keys() if k != "USDC"],
    ↪index=["estatistica_F", "p_valor"])

for coluna in levene_df:
    levene_df[coluna] = levene(acoes_df["USDC"], acoes_df[coluna])

levene_df
```

```
[7]:
```

	BITCOIN	CARDANO	LITECOIN	CHILIZ \
estatistica_F	3068.42429	1.557726e+02	1.791099e+03	5.163369e+02
p_valor	0.00000	2.021245e-34	1.601442e-276	2.389128e-101

	NANO
estatistica_F	7.629414e+02
p_valor	1.545937e-141

```
[8]: bartlett_df = pd.DataFrame(columns=[k for k in ativos.keys() if k != "USDC"],
    ↪index=["K_squared", "p_valor"])

for coluna in bartlett_df:
    bartlett_df[coluna] = bartlett(acoes_df["USDC"], acoes_df[coluna])

bartlett_df
```

```
[8]:
```

	BITCOIN	CARDANO	LITECOIN	CHILIZ	NANO
K_squared	23349.312833	2483.70839	12745.147401	1.183827e+03	8533.816943
p_valor	0.000000	0.00000	0.000000	1.996110e-259	0.000000

Com  $p < 0.05$  assumimos que médias dos grupos observados são iguais, ou seja, não há variância na média dos grupos.

$$H_0 = \mu_0 = \mu_1 = \mu_2 \dots \mu_p \quad (1)$$

### 0.2.3 Gráfico de Avaliação de Normalidade dos Resíduos

```
[9]: anova_df = pd.DataFrame(columns=[k for k in ativos.keys() if k != "USDC"],
    ↪index=["f_valor", "p_valor"])

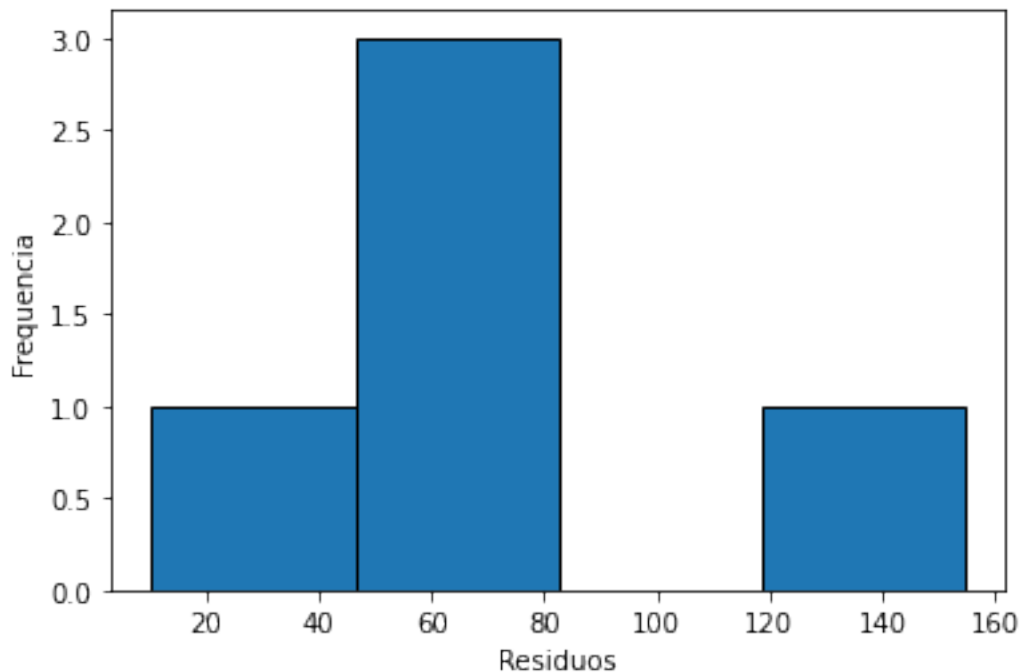
for coluna in anova_df:
    anova_df[coluna] = stats.f_oneway(acoes_df["USDC"], acoes_df[coluna])

anova_df
```

```
[9]:
```

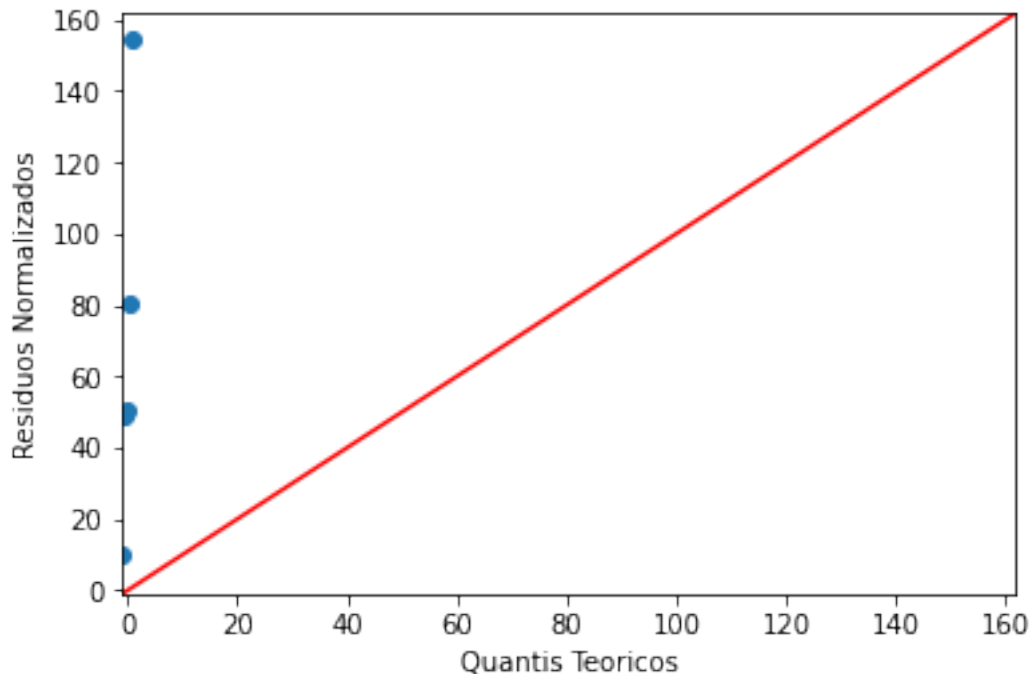
	BITCOIN	CARDANO	LITECOIN	CHILIZ	NANO
f_valor	2527.832856	1.087859e+02	2423.398567	23891.813156	6482.705866
p_valor	0.000000	8.293112e-25	0.000000	0.000000	0.000000

```
[10]: # histograma
plt.hist(np.sqrt(anova_df.iloc[0]), bins="auto", histtype="bar", ec="k")
plt.xlabel("Resíduos")
plt.ylabel("Frequencia")
plt.show()
```



Os resíduos dos grupos observados apresentam uma distribuição normal aproximada, ou seja, homocedasticidade das variâncias.

```
[11]: sm.qqplot(np.sqrt(anova_df.iloc[0]), line="45")
plt.xlabel("Quantis Teóricos")
plt.ylabel("Resíduos Normalizados")
plt.show()
```



Observamos uma violação da suposição levantada no gráfico anterior. Muito provavelmente pelo dado histórico do preço do criptoativo USDC ser contínuo.

#### 0.2.4 Tratamento Dos Dados Com Uma Distribuição Não Normal

Como falhamos em deduzir que os dados se distribuem de forma normal, é preciso realizar o teste de Mann Whitney, para checar se as amostras possuem a mesma distribuição (independente de qual seja).

**Teste de Mann Whitney** Estamos assumindo que as amostras são independentes, ou seja, para cada par  $x, y$ , assumimos que  $x$  é independente de  $y$ .

```
[40]: mann_whitney = pd.DataFrame(columns=[k for k in ativos.keys() if k != "USDC"],
    ↪ index=["u_test", "p_valor"])

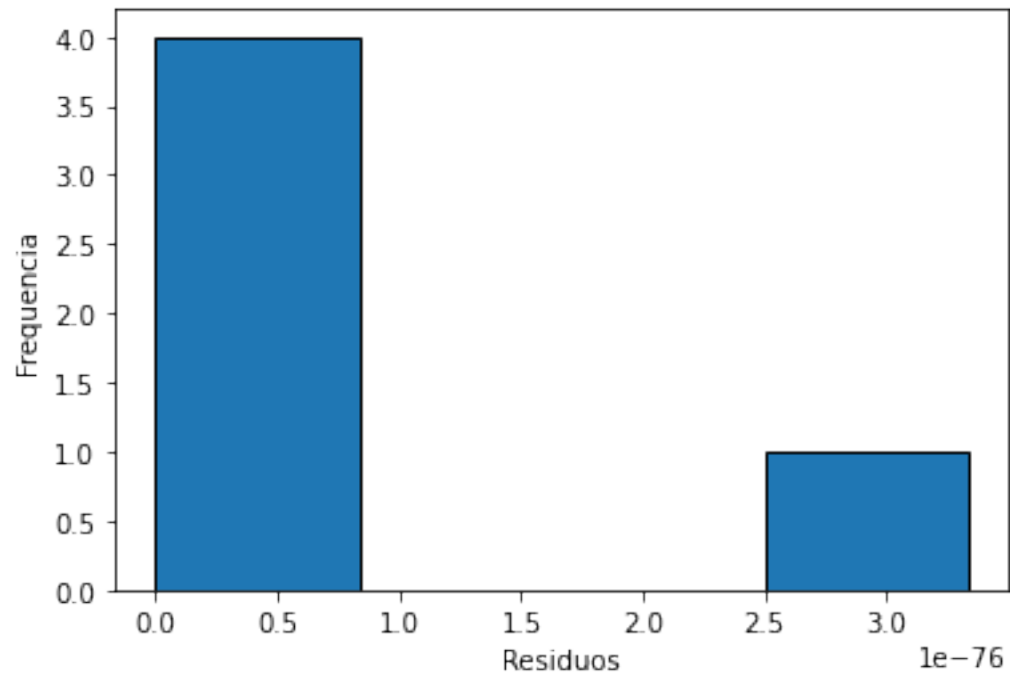
for coluna in mann_whitney:
    mann_whitney[coluna] = mannwhitneyu(acoef_df["USDC"], acoes_df[coluna],
    ↪ alternative="two-sided", method="asymptotic")

mann_whitney
```

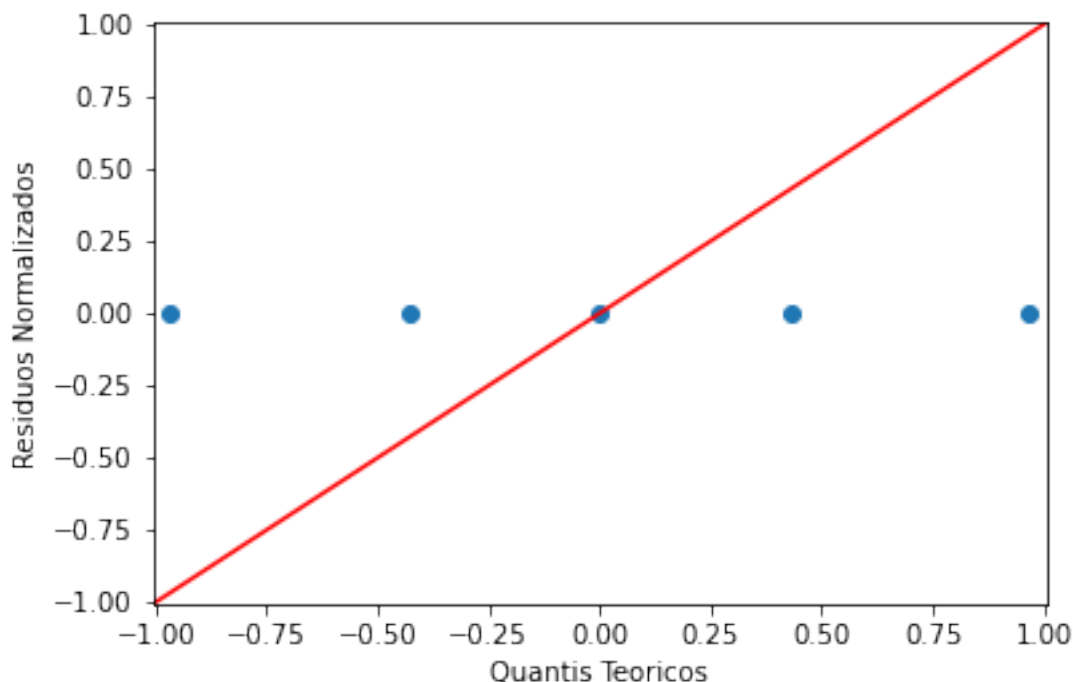
```
[40]:
```

	BITCOIN	CARDANO	LITECOIN	CHILIZ	NANO
u_test	0.0	1.459910e+05	0.0	910116.0	0.0
p_valor	0.0	1.110911e-151	0.0	0.0	0.0

```
[41]: # histograma
plt.hist(np.sqrt(mann_whitney.iloc[1]), bins="auto", histtype="bar", ec="k")
plt.xlabel("Residuos")
plt.ylabel("Frecuencia")
plt.show()
```



```
[42]: sm.qqplot(np.sqrt(mann_whitney.iloc[1]), line="45")
plt.xlabel("Quantis Teoricos")
plt.ylabel("Residuos Normalizados")
plt.show()
```



A partir da interpretação dos dados e dos gráficos, é possível concluir que não há uma diferença estatística significativa entre os dados.

**Teste da ANOVA Através do Teste H de Kruskal-Wallis** O teste H de Kruskal-Wallis testa a hipótese nula de que as medianas populacionais de todos os grupos são iguais. É uma versão não paramétrica da ANOVA. O teste funciona em 2 ou mais amostras independentes, que podem ter tamanhos diferentes. Observe que rejeitar a hipótese nula não indica qual dos grupos difere.

```
[36]: kruskall = pd.DataFrame(columns=[k for k in ativos.keys() if k != "USDC"],
    ↪ index=["statistic", "p_valor"])

for coluna in kruskall:
    kruskall[coluna] = kruskal(acoes_df["USDC"], acoes_df[coluna])

kruskall
```

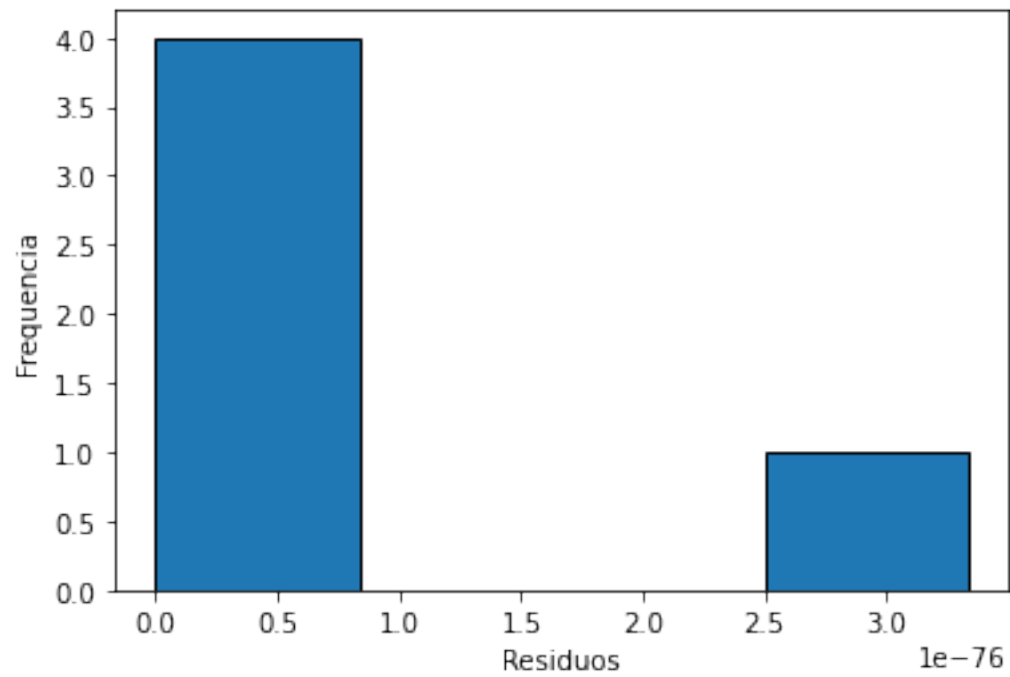
```
[36]:
```

	BITCOIN	CARDANO	LITECOIN	CHILIZ	NANO
statistic	1430.93207	6.881818e+02	1430.932068	1432.558138	1439.823699
p_valor	0.00000	1.110911e-151	0.000000	0.000000	0.000000

```
[43]: # histograma
plt.hist(np.sqrt(kruskall.iloc[1]), bins="auto", histtype="bar", ec="k")
plt.xlabel("Resíduos")
plt.ylabel("Frequencia")
```



```
plt.show()
```



```
[44]: sm.qqplot(np.sqrt(kruskall.iloc[1]), line="45")  
      plt.xlabel("Quantis Teoricos")  
      plt.ylabel("Residuos Normalizados")  
      plt.show()
```

