

```

"""
Importando as bibliotecas
"""

import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import matplotlib.ticker as mticker
from matplotlib.ticker import AutoMinorLocator, MaxNLocator
from matplotlib.font_manager import FontProperties

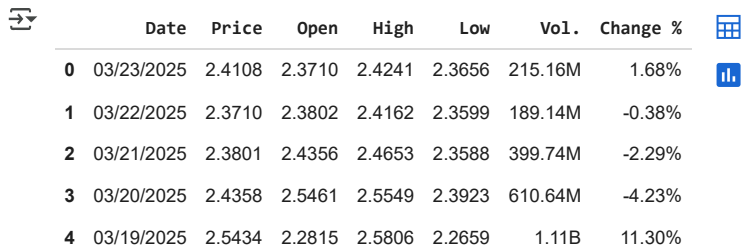
Normalização dos preços
"""

from sklearn.preprocessing import MinMaxScaler

"""
Importando os dados
"""

Data_XRP = pd.read_csv("/content/XRP Historical Data (1).csv")
Data_XRP.head()

```



	Date	Price	Open	High	Low	Vol.	Change %
0	03/23/2025	2.4108	2.3710	2.4241	2.3656	215.16M	1.68%
1	03/22/2025	2.3710	2.3802	2.4162	2.3599	189.14M	-0.38%
2	03/21/2025	2.3801	2.4356	2.4653	2.3588	399.74M	-2.29%
3	03/20/2025	2.4358	2.5461	2.5549	2.3923	610.64M	-4.23%
4	03/19/2025	2.5434	2.2815	2.5806	2.2659	1.11B	11.30%

Próximas etapas: [Gerar código com Data_XRP](#) [Ver gráficos recomendados](#) [New interactive sheet](#)

```

"""
Transformando a coluna "Date" em datetime
"""

Data_XRP["Date"] = pd.to_datetime(Data_XRP["Date"], format="%m/%d/%Y")

"""
Criação da primeira fonte de texto para colocar como fonte dos labels
"""

font1 = {"family": "serif", "weight": "bold", "color": "black", "size": 14}

"""
Criação da segunda fonte de texto para colocar como fonte da legenda
"""

font2 = FontProperties(family = "serif",
                      weight = "bold",
                      size = 14)

"""
Cria um "lugar" com size (9, 7) para alocar a figura
"""

fig, axs = plt.subplots(figsize = (16, 7))

"""
Plota um scatter entre o total de casos (cumulativo) e total de mortes (cumulativo)
"""

axs.plot(Data_XRP["Date"], Data_XRP["Price"], color = "orange")
axs.grid(False)

"""
Definindo a "grossura" e a cor do eixos
"""

for axis in ["left", "right", "top", "bottom"]:
    axs.spines[axis].set_linewidth(2)
    axs.spines[axis].set_color("black")

"""
Trabalha com os ticks do gráfico
"""

axs.xaxis.set_minor_locator(AutoMinorLocator())
axs.yaxis.set_minor_locator(AutoMinorLocator())
axs.tick_params(axis = "both", direction = "in", labelcolor = "black", labelsiz = 14, left = True, bottom = True, top = True, right = True)
axs.tick_params(which = "major", direction = "in", color = "black", length = 5.4, width = 2.5, left = True, bottom = True, top = True, right = True)
axs.tick_params(which = "minor", direction = "in", color = "black", length=4, width = 2, left = True, bottom = True, top = True, right = True)

"""
Descrição para cada eixo
"""

axs.set_xlabel("Data", fontdict = font1)
axs.set_ylabel("Price (US$)", fontdict = font1)

plt.rcParams["axes.labelweight"] = "bold" mostra em negrito os números nos eixos.

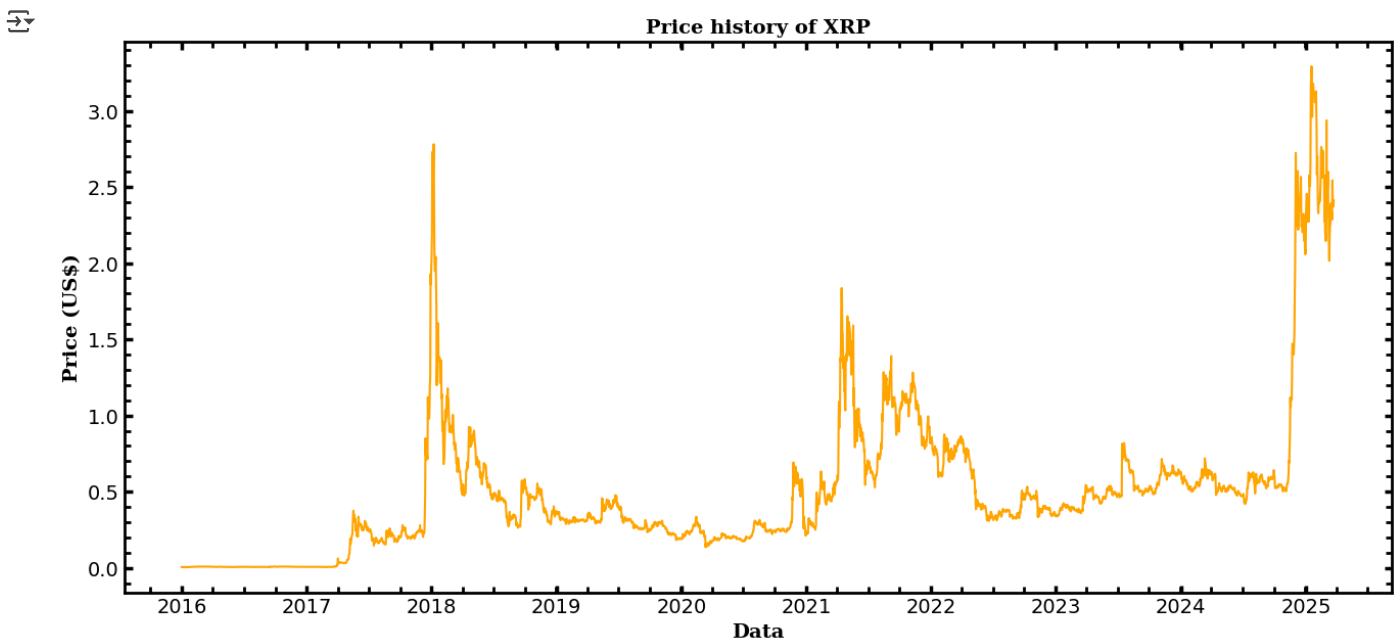
```

```
"""
plt.rcParams["axes.labelweight"] = "bold"
"""

Definindo um fundo branco para a imagem
"""
fig.patch.set_facecolor("white")
Cor_fundo = plt.gca()
Cor_fundo.set_facecolor("white")
Cor_fundo.patch.set_alpha(1)
"""

Título
"""
plt.title("Price history of XRP", fontdict=font1)
"""

Mostrar o gráfico
"""
plt.show()
```



```
Data_BTC = pd.read_csv("/content/Bitcoin Historical Data (1).csv")
Data_BTC.head()
```

	Date	Price	Open	High	Low	Vol.	Change %
0	03/23/2025	84,766.0	83,840.2	85,008.4	83,812.7	22.27K	1.10%
1	03/22/2025	83,840.5	84,076.6	84,526.3	83,688.2	20.84K	-0.28%
2	03/21/2025	84,073.3	84,208.7	84,831.0	83,215.9	42.80K	-0.16%
3	03/20/2025	84,208.1	86,841.3	87,419.0	83,656.1	69.86K	-3.03%
4	03/19/2025	86,841.0	82,719.1	86,990.6	82,557.5	84.61K	4.98%

Próximas etapas: [Gerar código com Data_BTC](#) [Ver gráficos recomendados](#) [New interactive sheet](#)

```
Data_BTC["Date"] = pd.to_datetime(Data_BTC["Date"], format="%m/%d/%Y")
```

```
Data_BTC["Price"] = Data_BTC["Price"].str.replace(",", "")
Data_BTC["Price"] = Data_BTC.Price.astype(float)
```

```
"""
Criação da primeira fonte de texto para colocar como fonte dos labels
"""
font1 = {"family": "serif", "weight": "bold", "color": "black", "size": 14}
"""
Criação da segunda fonte de texto para colocar como fonte da legenda
```

```

"""
font2 = FontProperties(family = "serif",
                      weight = "bold",
                      size = 14)
"""

Cria um "lugar" com size (9, 7) para alocar a figura
"""
fig, axs = plt.subplots(figsize = (16, 7))
"""

Plota um scatter entre o total de casos (cumulativo) e total de mortes (cumulativo)
"""
axs.plot(Data_BTC["Date"], Data_BTC["Price"], color = "orange")
axs.grid(False)
"""

Definindo a "grossura" e a cor do eixos
"""
for axis in ["left", "right", "top", "bottom"]:
    axs.spines[axis].set_linewidth(2)
    axs.spines[axis].set_color("black")
"""

Trabalha com os ticks do gráfico
"""
axs.xaxis.set_minor_locator(AutoMinorLocator())
axs.yaxis.set_minor_locator(AutoMinorLocator())
axs.tick_params(axis = "both", direction = "in", labelcolor = "black", labelsiz = 14, left = True, bottom = True, top = True, right = True)
axs.tick_params(which = "major", direction = "in", color = "black", length = 5.4, width = 2.5, left = True, bottom = True, top = True, right = True)
axs.tick_params(which = "minor", direction = "in", color = "black", length=4, width = 2, left = True, bottom = True, top = True, right = True)
"""

Descrição para cada eixo
"""
axs.set_xlabel("Data", fontdict = font1)
axs.set_ylabel("Price (US$)", fontdict = font1)
"""

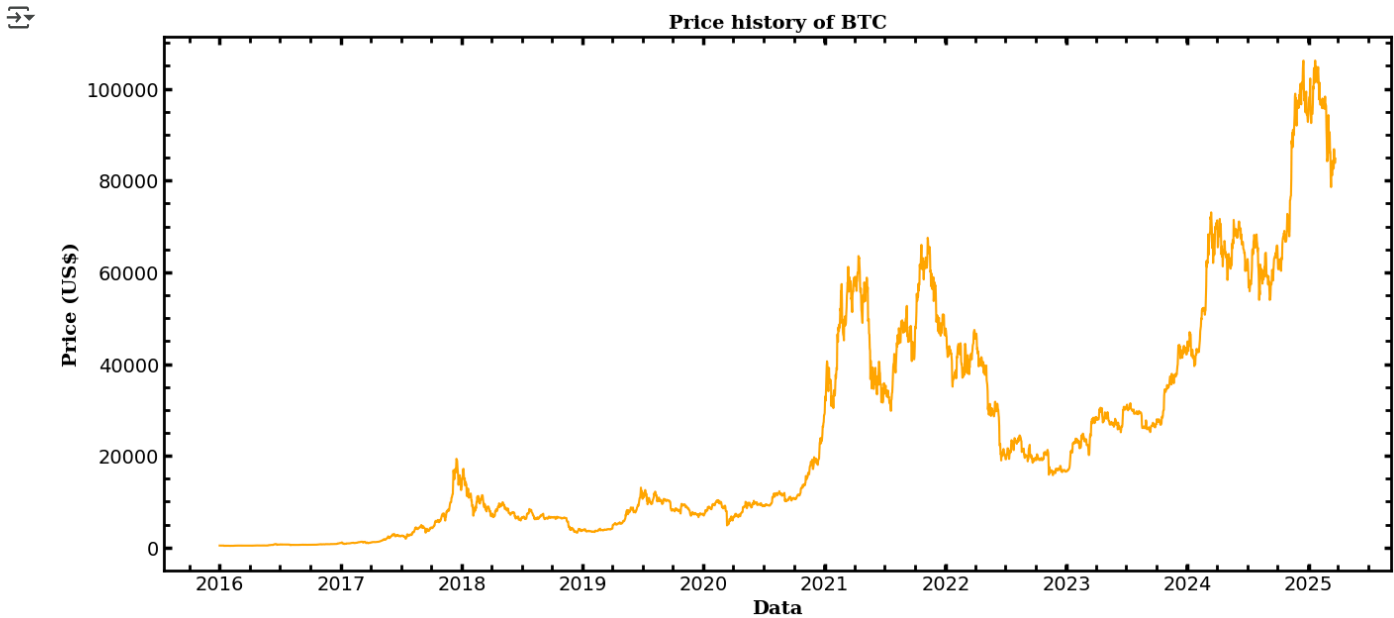
plt.rcParams["axes.labelweight"] = "bold" mostra em negrito os números nos eixos.
"""
plt.rcParams["axes.labelweight"] = "bold"
"""

Definindo um fundo branco para a imagem
"""
fig.patch.set_facecolor("white")
Cor_fundo = plt.gca()
Cor_fundo.set_facecolor("white")
Cor_fundo.patch.set_alpha(1)
"""

Título
"""
plt.title("Price history of BTC", fontdict=font1)
"""

Mostrar o gráfico
"""
plt.show()

```



```
Norm = MinMaxScaler()
Data_BTC["Price"] = Data_BTC["Price"].values.reshape(-1, 1)
Data_BTC["Price"] = Norm.fit_transform(Data_BTC[["Price"]])
Data_BTC
```

↗

	Date	Price	Open	High	Low	Vol.	Change %	
0	2025-03-23	0.797794	83,840.2	85,008.4	83,812.7	22.27K	1.10%	
1	2025-03-22	0.789046	84,076.6	84,526.3	83,688.2	20.84K	-0.28%	
2	2025-03-21	0.791246	84,208.7	84,831.0	83,215.9	42.80K	-0.16%	
3	2025-03-20	0.792520	86,841.3	87,419.0	83,656.1	69.86K	-3.03%	
4	2025-03-19	0.817409	82,719.1	86,990.6	82,557.5	84.61K	4.98%	
...	
3364	2016-01-05	0.000597	433.3	435.3	428.9	45.03K	-0.49%	
3365	2016-01-04	0.000617	430.7	435.3	428.6	53.01K	0.61%	
3366	2016-01-03	0.000593	433.7	434.1	423.1	54.83K	-0.70%	
3367	2016-01-02	0.000621	434.0	437.4	430.7	33.57K	-0.06%	
3368	2016-01-01	0.000624	430.0	438.0	425.9	46.97K	0.94%	

3369 rows × 7 columns

Próximas etapas:

[Gerar código com Data_BTC](#)

☒ Ver gráficos recomendados

[New interactive sheet](#)

```
Data_XRP["Price"] = Data_XRP["Price"].values.reshape(-1, 1)
Data_XRP["Price"] = Norm.fit_transform(Data_XRP[["Price"]])
Data_XRP
```

	Date	Price	Open	High	Low	Vol.	Change %	
0	2025-03-23	0.732129	2.3710	2.4241	2.3656	215.16M	1.68%	
1	2025-03-22	0.720022	2.3802	2.4162	2.3599	189.14M	-0.38%	
2	2025-03-21	0.722790	2.4356	2.4653	2.3588	399.74M	-2.29%	
3	2025-03-20	0.739734	2.5461	2.5549	2.3923	610.64M	-4.23%	
4	2025-03-19	0.772465	2.2815	2.5806	2.2659	1.11B	11.30%	
...	
3364	2016-01-05	0.000304	0.0051	0.0054	0.0050	NaN	-1.96%	
3365	2016-01-04	0.000335	0.0052	0.0054	0.0051	NaN	-1.92%	
3366	2016-01-03	0.000365	0.0051	0.0054	0.0050	0.02K	1.46%	
3367	2016-01-02	0.000335	0.0051	0.0051	0.0050	0.50K	-6.82%	
3368	2016-01-01	0.000456	0.0052	0.0055	0.0051	NaN	5.67%	

3369 rows × 7 columns

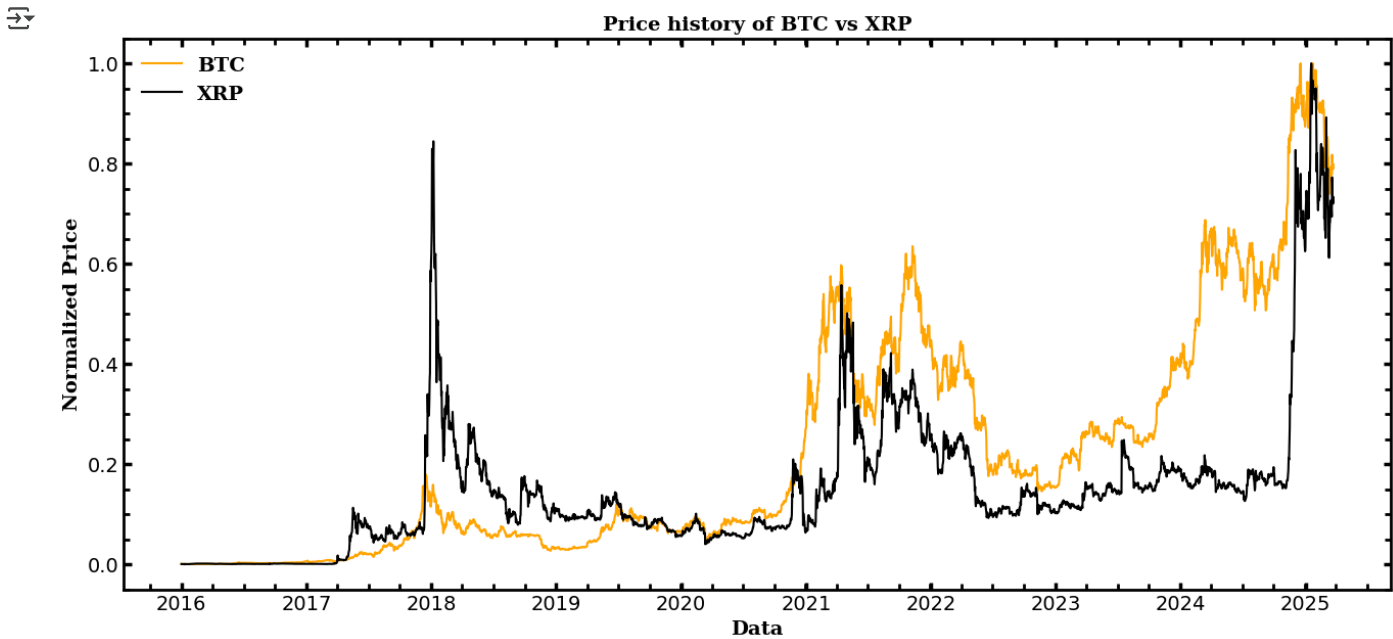
Próximas etapas: [Gerar código com Data_XRP](#) [Ver gráficos recomendados](#) [New interactive sheet](#)

```

"""
Criação da primeira fonte de texto para colocar como fonte dos labels
"""
font1 = {"family": "serif", "weight": "bold", "color": "black", "size": 14}
"""
Criação da segunda fonte de texto para colocar como fonte da legenda
"""
font2 = FontProperties(family = "serif",
                      weight = "bold",
                      size = 14)
"""
Cria um "lugar" com size (9, 7) para alocar a figura
"""
fig, axs = plt.subplots(figsize = (16, 7))
"""
Plota um scatter entre o total de casos (cumulativo) e total de mortes (cumulativo)
"""
axs.plot(Data_BTC["Date"], Data_BTC["Price"], color = "orange")
axs.plot(Data_XRP["Date"], Data_XRP["Price"], color = "black")
axs.grid(False)
"""
Definindo a "grossura" e a cor do eixos
"""
for axis in ["left", "right", "top", "bottom"]:
    axs.spines[axis].set_linewidth(2)
    axs.spines[axis].set_color("black")
"""
Trabalha com os ticks do gráfico
"""
axs.xaxis.set_minor_locator(AutoMinorLocator())
axs.yaxis.set_minor_locator(AutoMinorLocator())
axs.tick_params(axis = "both", direction = "in", labelcolor = "black", labelsize = 14, left = True, bottom = True, top = True, right = True)
axs.tick_params(which = "major", direction = "in", color = "black", length = 5.4, width = 2.5, left = True, bottom = True, top = True, right = True)
axs.tick_params(which = "minor", direction = "in", color = "black", length=4, width = 2, left = True, bottom = True, top = True, right = True)
"""
Descrição para cada eixo
"""
axs.set_xlabel("Data", fontdict = font1)
axs.set_ylabel("Normalized Price", fontdict = font1)
"""
plt.rcParams["axes.labelweight"] = "bold" mostra em negrito os números nos eixos.
"""
plt.rcParams["axes.labelweight"] = "bold"
"""
Definindo um fundo branco para a imagem
"""
fig.patch.set_facecolor("white")
Cor_fundo = plt.gca()
Cor_fundo.set_facecolor("white")
Cor_fundo.patch.set_alpha(1)
"""
Título
"""
plt.title("Price history of BTC vs XRP", fontdict=font1)
"""
Mostrar o gráfico

```

```
"""
"""
Legenda
"""
plt.legend(["BTC", "XRP"], prop = font2, frameon=False)
"""
Mostrar o gráfico
"""
plt.show()
```



```
Data_ETH = pd.read_csv("/content/Ethereum Historical Data.csv")
Data_ETH
```

	Date	Price	Open	High	Low	Vol.	Change %	
0	03/23/2025	2,009.14	1,980.78	2,019.59	1,977.07	249.72K	1.43%	
1	03/22/2025	1,980.80	1,965.74	2,005.80	1,965.45	185.64K	0.77%	
2	03/21/2025	1,965.58	1,983.94	1,996.73	1,938.55	291.89K	-0.91%	
3	03/20/2025	1,983.57	2,056.11	2,066.45	1,952.95	433.71K	-3.55%	
4	03/19/2025	2,056.62	1,931.79	2,069.43	1,927.77	743.84K	6.42%	
...	
3296	03/14/2016	12.50	15.07	15.07	11.40	92.18K	-17.05%	
3297	03/13/2016	15.07	12.92	15.07	12.92	1.30K	16.64%	
3298	03/12/2016	12.92	11.95	13.45	11.95	0.83K	8.12%	
3299	03/11/2016	11.95	11.75	11.95	11.75	0.18K	1.70%	
3300	03/10/2016	11.75	11.20	11.85	11.07	0.00K	4.91%	

3301 rows × 7 columns

Próximas etapas: [Gerar código com Data_ETH](#) [Ver gráficos recomendados](#) [New interactive sheet](#)

```
Data_ETH["Date"] = pd.to_datetime(Data_ETH["Date"], format="%m/%d/%Y")
```

```
Data_ETH["Price"] = Data_ETH["Price"].str.replace(",", "")
Data_ETH["Price"] = Data_ETH.Price.astype(float)
```

```
Data_ETH["Price"] = Data_ETH["Price"].values.reshape(-1, 1)
Data_ETH["Price"] = Norm.fit_transform(Data_ETH[["Price"]])
Data_ETH
```

	Date	Price	Open	High	Low	Vol.	Change %
0	2025-03-23	0.417029	1,980.78	2,019.59	1,977.07	249.72K	1.43%
1	2025-03-22	0.411127	1,965.74	2,005.80	1,965.45	185.64K	0.77%
2	2025-03-21	0.407957	1,983.94	1,996.73	1,938.55	291.89K	-0.91%
3	2025-03-20	0.411704	2,056.11	2,066.45	1,952.95	433.71K	-3.55%
4	2025-03-19	0.426917	1,931.79	2,069.43	1,927.77	743.84K	6.42%
...
3296	2016-03-14	0.001208	15.07	15.07	11.40	92.18K	-17.05%
3297	2016-03-13	0.001743	12.92	15.07	12.92	1.30K	16.64%
3298	2016-03-12	0.001295	11.95	13.45	11.95	0.83K	8.12%
3299	2016-03-11	0.001093	11.75	11.95	11.75	0.18K	1.70%
3300	2016-03-10	0.001052	11.20	11.85	11.07	0.00K	4.91%

3301 rows × 7 columns

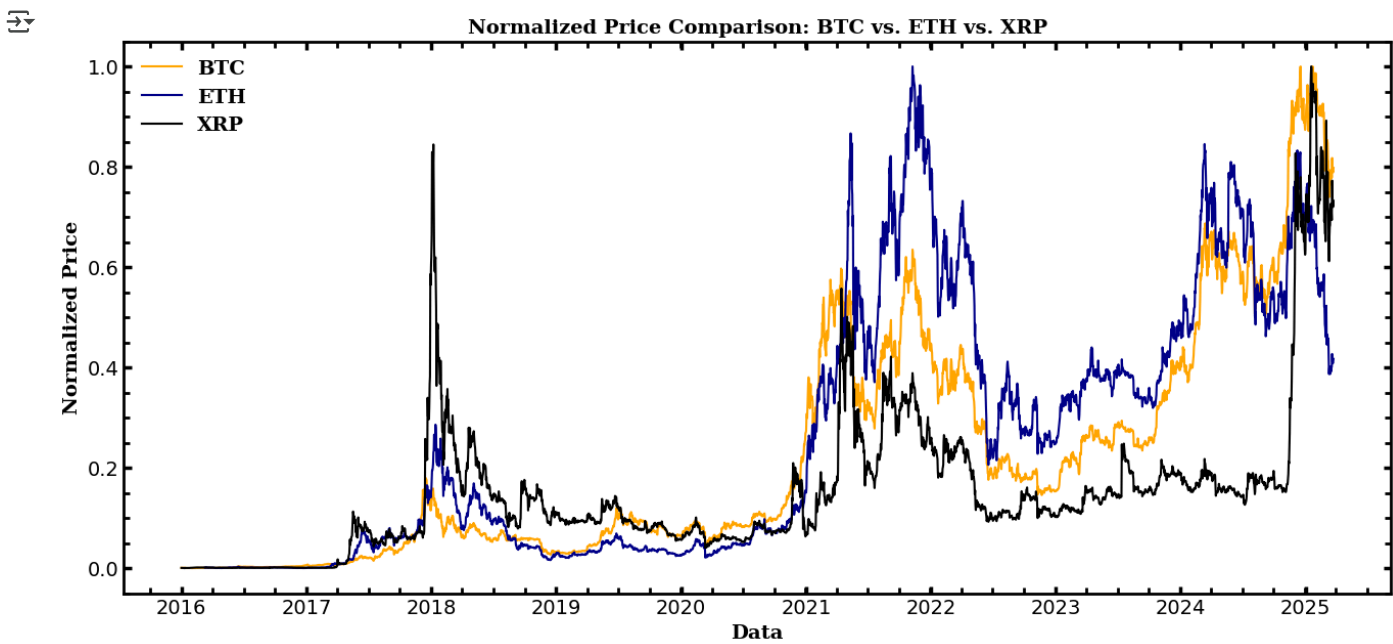
Próximas etapas: [Gerar código com Data_ETH](#) [Ver gráficos recomendados](#) [New interactive sheet](#)

```

"""
Criação da primeira fonte de texto para colocar como fonte dos labels
"""
font1 = {"family": "serif", "weight": "bold", "color": "black", "size": 14}
"""
Criação da segunda fonte de texto para colocar como fonte da legenda
"""
font2 = FontProperties(family = "serif",
                      weight = "bold",
                      size = 14)
"""
Cria um "lugar" com size (9, 7) para alocar a figura
"""
fig, axs = plt.subplots(figsize = (16, 7))
"""
Plota um scatter entre o total de casos (cumulativo) e total de mortes (cumulativo)
"""
axs.plot(Data_BTC["Date"], Data_BTC["Price"], color = "orange")
axs.plot(Data_ETH["Date"], Data_ETH["Price"], color = "darkblue")
axs.plot(Data_XRP["Date"], Data_XRP["Price"], color = "black")
axs.grid(False)
"""
Definindo a "grossura" e a cor do eixos
"""
for axis in ["left", "right", "top", "bottom"]:
    axs.spines[axis].set_linewidth(2)
    axs.spines[axis].set_color("black")
"""
Trabalha com os ticks do gráfico
"""
axs.xaxis.set_minor_locator(AutoMinorLocator())
axs.yaxis.set_minor_locator(AutoMinorLocator())
axs.tick_params(axis = "both", direction = "in", labelcolor = "black", labelsiz = 14, left = True, bottom = True, top = True, right = True)
axs.tick_params(which = "major", direction = "in", color = "black", length = 5.4, width = 2.5, left = True, bottom = True, top = True, right = True)
axs.tick_params(which = "minor", direction = "in", color = "black", length=4, width = 2, left = True, bottom = True, top = True, right = True)
"""
Descrição para cada eixo
"""
axs.set_xlabel("Data", fontdict = font1)
axs.set_ylabel("Normalized Price", fontdict = font1)
"""
plt.rcParams["axes.labelweight"] = "bold" mostra em negrito os números nos eixos.
"""
plt.rcParams["axes.labelweight"] = "bold"
"""
Definindo um fundo branco para a imagem
"""
fig.patch.set_facecolor("white")
Cor_fundo = plt.gca()
Cor_fundo.set_facecolor("white")
Cor_fundo.patch.set_alpha(1)
"""
Título
"""
plt.title("Normalized Price Comparison: BTC vs. ETH vs. XRP", fontdict=font1)
"""

```

```
Mostrar o gráfico
"""
"""
Legenda
"""
plt.legend(["BTC", "ETH", "XRP"], prop = font2, frameon=False)
"""
Mostrar o gráfico
"""
plt.show()
```



```
Data_ADA = pd.read_csv("/content/Cardano Historical Data.csv")
Data_ADA
```

	Date	Price	Open	High	Low	Vol.	Change %
0	03/23/2025	0.7112	0.7014	0.7158	0.7008	141.88M	1.40%
1	03/22/2025	0.7014	0.7050	0.7160	0.6975	110.49M	-0.49%
2	03/21/2025	0.7048	0.7179	0.7273	0.7007	159.73M	-1.81%
3	03/20/2025	0.7178	0.7437	0.7551	0.7125	233.04M	-3.47%
4	03/19/2025	0.7437	0.7018	0.7467	0.7006	281.11M	5.96%
...
2634	01/04/2018	1.1043	1.1000	1.3500	1.0500	19.60M	2.25%
2635	01/03/2018	1.0800	0.9782	1.0900	0.8882	20.23M	41.73%
2636	01/02/2018	0.7620	0.7022	0.8000	0.6750	10.97M	8.52%
2637	01/01/2018	0.7022	0.7100	0.7150	0.6320	17.95M	-1.10%
2638	12/31/2017	0.7100	0.5900	0.7400	0.5900	17.22M	20.34%

2639 rows x 7 columns

Próximas etapas: [Gerar código com Data_ADA](#) [Ver gráficos recomendados](#) [New interactive sheet](#)

```
Data_ADA["Date"] = pd.to_datetime(Data_ADA["Date"], format="%m/%d/%Y")

Data_ADA["Price"] = Data_ADA["Price"].values.reshape(-1, 1)
Data_ADA["Price"] = Norm.fit_transform(Data_ADA[["Price"]])
Data_ADA
```


	Date	Price	Open	High	Low	Vol.	Change %	
0	2025-03-23	0.233855	0.7014	0.7158	0.7008	141.88M	1.40%	
1	2025-03-22	0.230523	0.7050	0.7160	0.6975	110.49M	-0.49%	
2	2025-03-21	0.231679	0.7179	0.7273	0.7007	159.73M	-1.81%	
3	2025-03-20	0.236098	0.7437	0.7551	0.7125	233.04M	-3.47%	
4	2025-03-19	0.244901	0.7018	0.7467	0.7006	281.11M	5.96%	
...	
2634	2018-01-04	0.367471	1.1000	1.3500	1.0500	19.60M	2.25%	
2635	2018-01-03	0.359211	0.9782	1.0900	0.8882	20.23M	41.73%	
2636	2018-01-02	0.251122	0.7022	0.8000	0.6750	10.97M	8.52%	
2637	2018-01-01	0.230795	0.7100	0.7150	0.6320	17.95M	-1.10%	
2638	2017-12-31	0.233447	0.5900	0.7400	0.5900	17.22M	20.34%	

2639 rows × 7 columns

Próximas etapas: [Gerar código com Data_ADA](#) [Ver gráficos recomendados](#) [New interactive sheet](#)

```

"""
Criação da primeira fonte de texto para colocar como fonte dos labels
"""
font1 = {"family": "serif", "weight": "bold", "color": "black", "size": 14}
"""
Criação da segunda fonte de texto para colocar como fonte da legenda
"""
font2 = FontProperties(family = "serif",
                      weight = "bold",
                      size = 14)
"""
Cria um "lugar" com size (9, 7) para alocar a figura
"""
fig, axs = plt.subplots(figsize = (16, 7))
"""
Plota um scatter entre o total de casos (cumulativo) e total de mortes (cumulativo)
"""
axs.plot(Data_BTC["Date"], Data_BTC["Price"], color = "orange")
axs.plot(Data_ETH["Date"], Data_ETH["Price"], color = "darkblue")
axs.plot(Data_XRP["Date"], Data_XRP["Price"], color = "black")
axs.plot(Data_ADA["Date"], Data_ADA["Price"], color = "red")
axs.grid(False)
"""
Definindo a "grossura" e a cor do eixos
"""
for axis in ["left", "right", "top", "bottom"]:
    axs.spines[axis].set_linewidth(2)
    axs.spines[axis].set_color("black")
"""
Trabalha com os ticks do gráfico
"""
axs.xaxis.set_minor_locator(AutoMinorLocator())
axs.yaxis.set_minor_locator(AutoMinorLocator())
axs.tick_params(axis = "both", direction = "in", labelcolor = "black", labelsiz = 14, left = True, bottom = True, top = True, right = True)
axs.tick_params(which = "major", direction = "in", color = "black", length = 5.4, width = 2.5, left = True, bottom = True, top = True, right = True)
axs.tick_params(which = "minor", direction = "in", color = "black", length=4, width = 2, left = True, bottom = True, top = True, right = True)
"""
Descrição para cada eixo
"""
axs.set_xlabel("Data", fontdict = font1)
axs.set_ylabel("Normalized Price", fontdict = font1)
"""
plt.rcParams["axes.labelweight"] = "bold" mostra em negrito os números nos eixos.
"""
plt.rcParams["axes.labelweight"] = "bold"
"""
Definindo um fundo branco para a imagem
"""
fig.patch.set_facecolor("white")
Cor_fundo = plt.gca()
Cor_fundo.set_facecolor("white")
Cor_fundo.patch.set_alpha(1)
"""
Título
"""
plt.title("Normalized Price Comparison: BTC vs. ETH vs. XRP vs. ADA", fontdict=font1)

```

```

"""
Mostrar o gráfico
"""

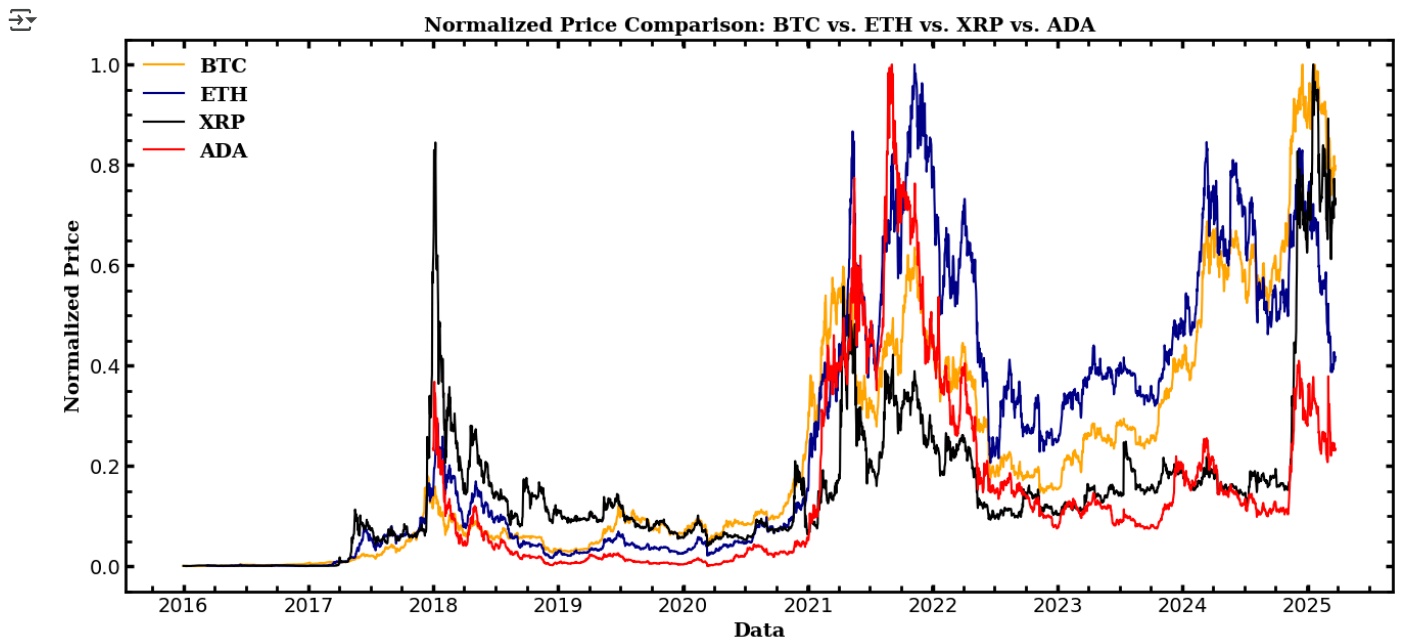
Legenda
"""

plt.legend(["BTC", "ETH", "XRP", "ADA"], prop = font2, frameon=False)
"""

Mostrar o gráfico
"""

plt.show()

```



```

"""
Criação da primeira fonte de texto para colocar como fonte dos labels
"""

font1 = {"family": "serif", "weight": "bold", "color": "black", "size": 14}
"""

Criação da segunda fonte de texto para colocar como fonte da legenda
"""

font2 = FontProperties(family = "serif",
                      weight = "bold",
                      size = 14)
"""

Cria um "lugar" com size (9, 7) para alocar a figura
"""

fig, axs = plt.subplots(figsize = (16, 7))
"""

Plota um scatter entre o total de casos (cumulativo) e total de mortes (cumulativo)
"""

axs.plot(Data_BTC["Date"], Data_BTC["Price"], color = "orange")
axs.plot(Data_ADA["Date"], Data_ADA["Price"], color = "red")
axs.grid(False)
"""

Definindo a "grossura" e a cor do eixos
"""

for axis in ["left", "right", "top", "bottom"]:
    axs.spines[axis].set_linewidth(2)
    axs.spines[axis].set_color("black")
"""

Trabalha com os ticks do gráfico
"""

axs.xaxis.set_minor_locator(AutoMinorLocator())
axs.yaxis.set_minor_locator(AutoMinorLocator())
axs.tick_params(axis = "both", direction = "in", labelcolor = "black", labelsiz = 14, left = True, bottom = True, top = True, right = False)
axs.tick_params(which = "major", direction = "in", color = "black", length = 5.4, width = 2.5, left = True, bottom = True, top = True, right = False)
axs.tick_params(which = "minor", direction = "in", color = "black", length=4, width = 2, left = True, bottom = True, top = True, right = False)
"""

Descrição para cada eixo

```

```

"""
axs.set_xlabel("Data", fontdict = font1)
axs.set_ylabel("Normalized Price", fontdict = font1)
"""

plt.rcParams["axes.labelweight"] = "bold" mostra em negrito os números nos eixos.
"""

plt.rcParams["axes.labelweight"] = "bold"
"""

Definindo um fundo branco para a imagem
"""

fig.patch.set_facecolor("white")
Cor_fundo = plt.gca()
Cor_fundo.set_facecolor("white")
Cor_fundo.patch.set_alpha(1)
"""

Título
"""

plt.title("Normalized Price Comparison: BTC vs. ADA", fontdict=font1)
"""

Mostrar o gráfico
"""

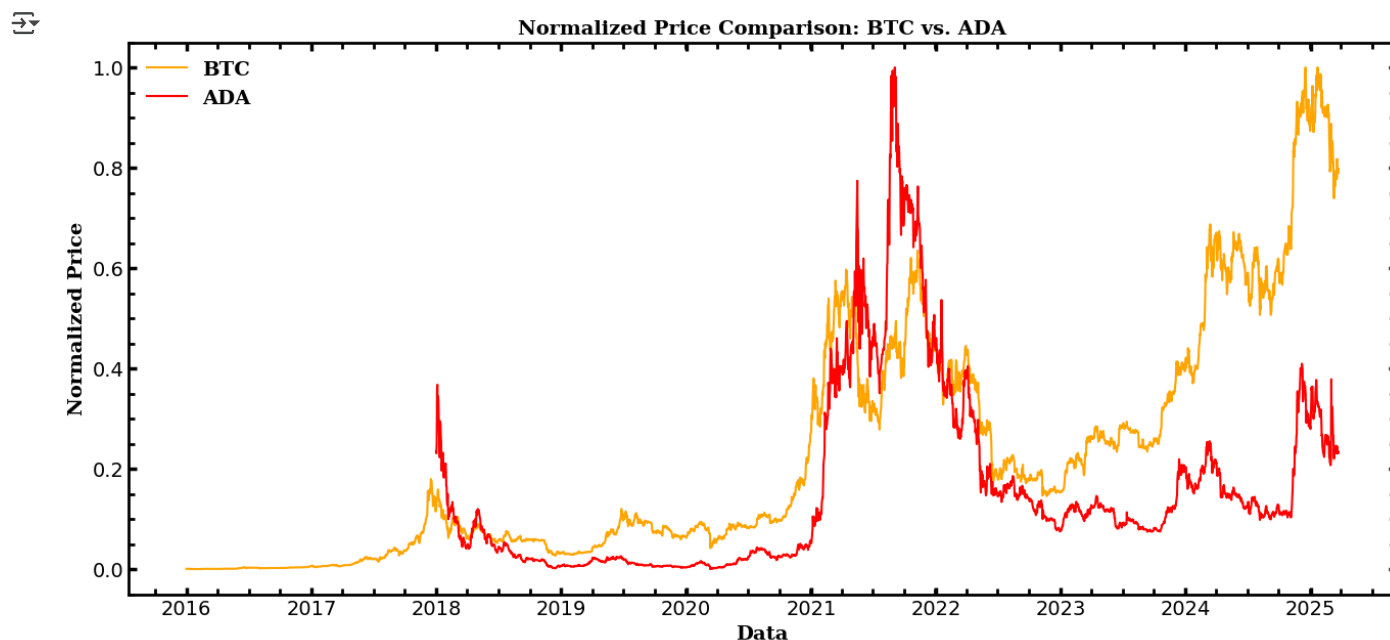
Legenda
"""

plt.legend(["BTC", "ADA"], prop = font2, frameon=False)
"""

Mostrar o gráfico
"""

plt.show()

```



```

"""
Criação da primeira fonte de texto para colocar como fonte dos labels
"""

font1 = {"family": "serif", "weight": "bold", "color": "black", "size": 14}
"""

Criação da segunda fonte de texto para colocar como fonte da legenda
"""

font2 = FontProperties(family = "serif",
                      weight = "bold",
                      size = 14)
"""

Cria um "lugar" com size (9, 7) para alocar a figura
"""

fig, axs = plt.subplots(figsize = (16, 7))
"""

Plota um scatter entre o total de casos (cumulativo) e total de mortes (cumulativo)
"""

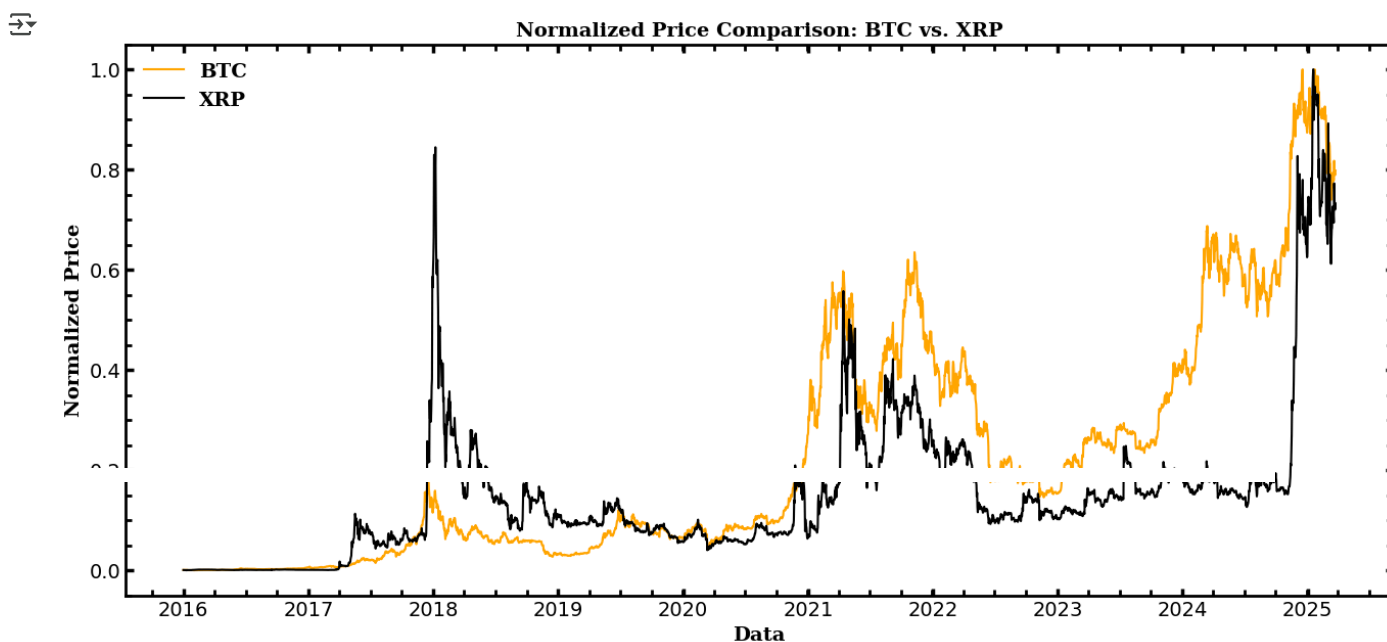
axs.plot(Data_BTC["Date"], Data_BTC["Price"], color = "orange")

```

```

axs.plot(Data_XRP["Date"], Data_XRP["Price"], color = "black")
axs.grid(False)
"""
Definindo a "grossura" e a cor do eixos
"""
for axis in ["left", "right", "top", "bottom"]:
    axs.spines[axis].set_linewidth(2)
    axs.spines[axis].set_color("black")
"""
Trabalha com os ticks do gráfico
"""
axs.xaxis.set_minor_locator(AutoMinorLocator())
axs.yaxis.set_minor_locator(AutoMinorLocator())
axs.tick_params(axis = "both", direction = "in", labelcolor = "black", labelsiz = 14, left = True, bottom = True, top = True, right = True)
axs.tick_params(which = "major", direction = "in", color = "black", length = 5.4, width = 2.5, left = True, bottom = True, top = True, right = True)
axs.tick_params(which = "minor", direction = "in", color = "black", length=4, width = 2, left = True, bottom = True, top = True, right = True)
"""
Descrição para cada eixo
"""
axs.set_xlabel("Data", fontdict = font1)
axs.set_ylabel("Normalized Price", fontdict = font1)
"""
plt.rcParams["axes.labelweight"] = "bold" mostra em negrito os números nos eixos.
"""
plt.rcParams["axes.labelweight"] = "bold"
"""
Definindo um fundo branco para a imagem
"""
fig.patch.set_facecolor("white")
Cor_fundo = plt.gca()
Cor_fundo.set_facecolor("white")
Cor_fundo.patch.set_alpha(1)
"""
Título
"""
plt.title("Normalized Price Comparison: BTC vs. XRP", fontdict=font1)
"""
Mostrar o gráfico
"""
"""
Legenda
"""
plt.legend(["BTC", "XRP"], prop = font2, frameon=False)
"""
Mostrar o gráfico
"""
plt.show()

```



```

Criação da primeira fonte de texto para colocar como fonte dos labels
"""

font1 = {"family": "serif", "weight": "bold", "color": "black", "size": 14}
"""

Criação da segunda fonte de texto para colocar como fonte da legenda
"""

font2 = FontProperties(family = "serif",
                      weight = "bold",
                      size = 14)
"""

Cria um "lugar" com size (9, 7) para alocar a figura
"""

fig, axs = plt.subplots(figsize = (16, 7))
"""

Plota um scatter entre o total de casos (cumulativo) e total de mortes (cumulativo)
"""

axs.plot(Data_BTC["Date"], Data_BTC["Price"], color = "orange")
axs.plot(Data_ETH["Date"], Data_ETH["Price"], color = "darkblue")
axs.grid(False)
"""

Definindo a "grossura" e a cor do eixos
"""

for axis in ["left", "right", "top", "bottom"]:
    axs.spines[axis].set_linewidth(2)
    axs.spines[axis].set_color("black")
"""

Trabalha com os ticks do gráfico
"""

axs.xaxis.set_minor_locator(AutoMinorLocator())
axs.yaxis.set_minor_locator(AutoMinorLocator())
axs.tick_params(axis = "both", direction = "in", labelcolor = "black", labelsize = 14, left = True, bottom = True, top = True, right = Tr
axs.tick_params(which = "major", direction = "in", color = "black", length = 5.4, width = 2.5, left = True, bottom = True, top = True, ri
axs.tick_params(which = "minor", direction = "in", color = "black", length=4, width = 2, left = True, bottom = True, top = True, right =
"""

Descrição para cada eixo
"""

axs.set_xlabel("Data", fontdict = font1)
axs.set_ylabel("Normalized Price", fontdict = font1)
"""

plt.rcParams["axes.labelweight"] = "bold" mostra em negrito os números nos eixos.
"""

plt.rcParams["axes.labelweight"] = "bold"
"""

Definindo um fundo branco para a imagem
"""

fig.patch.set_facecolor("white")
Cor_fundo = plt.gca()
Cor_fundo.set_facecolor("white")
Cor_fundo.patch.set_alpha(1)
"""

Título
"""

plt.title("Normalized Price Comparison: BTC vs. ETH", fontdict=font1)
"""

Mostrar o gráfico
"""

"""

Legenda
"""

plt.legend(["BTC", "ETH"], prop = font2, frameon=False)
"""

Mostrar o gráfico
"""

- - - - -

```