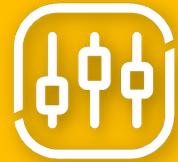
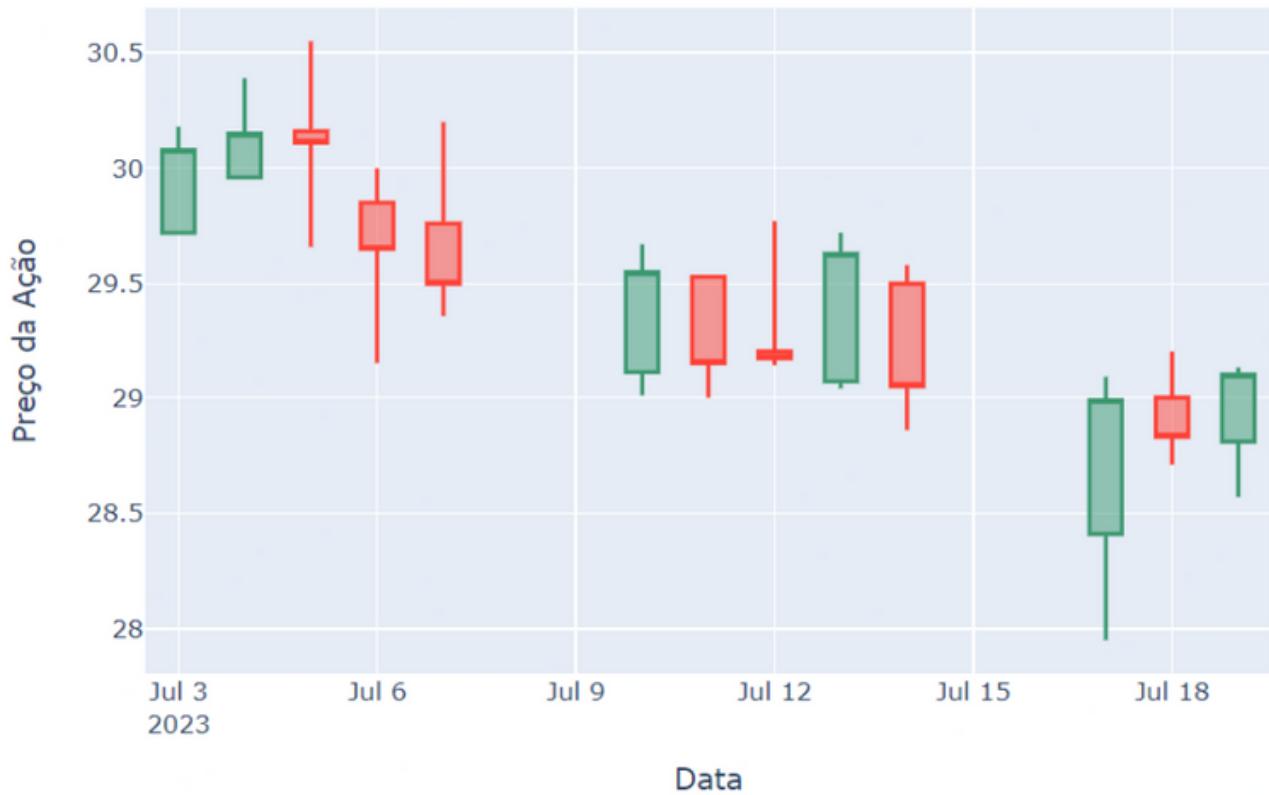


DATAVIZ



Gráficos para mercado financeiro

Variação de Preços da Ação PETR4.SA



INTRODUÇÃO



A visualização de dados é uma ferramenta essencial para analisar, compreender e comunicar informações complexas no mercado financeiro. Em um setor onde grandes volumes de dados são gerados diariamente, a visualização desempenha um papel fundamental na tomada de decisões informadas, identificação de padrões, detecção de tendências e avaliação de riscos.

No mercado financeiro, a quantidade de informações disponíveis pode ser avassaladora, incluindo cotações de ações, índices de mercado, dados econômicos, taxas de juros, preços de commodities, indicadores macroeconômicos, entre outros. Para analisar esses dados de forma eficiente, é necessário transformar números e estatísticas em representações visuais claras e concisas.

INTRODUÇÃO



Aqui estão algumas razões pelas quais a visualização de dados é crucial para o mercado financeiro:

- **Detecção de padrões e anomalias:** As visualizações possibilitam a identificação de padrões complexos e anomalias ocultas nos dados financeiros, fornecendo insights valiosos para a formulação de estratégias de investimento, gestão de riscos e otimização de portfólio.
- **Comunicação eficaz:** Gráficos e infográficos tornam as informações financeiras mais acessíveis e compreensíveis para todas as partes interessadas;

INTRODUÇÃO



Aqui estão algumas razões pelas quais a visualização de dados é crucial para o mercado financeiro:

- **Análise comparativa:** Visualizações permitem a comparação rápida de diferentes ativos, setores ou índices, ajudando a identificar o desempenho relativo e a avaliar a diversificação de portfólio.
- **Previsões e tendências:** Gráficos interativos e visualizações dinâmicas facilitam a análise de séries temporais, permitindo a identificação de tendências e padrões que podem ajudar na previsão de movimentos futuros do mercado.
- **Monitoramento de riscos:** Visualizações podem ajudar a monitorar o risco de investimentos e a exposição a volatilidades do mercado, auxiliando na proteção de ativos e na mitigação de riscos.

MATPLOTLIB



A biblioteca Matplotlib é uma das ferramentas mais poderosas e amplamente utilizadas em Python para criação de gráficos e visualizações. Ela oferece uma vasta gama de recursos e opções de personalização, tornando-a ideal para representar dados complexos no mercado financeiro. Através do Matplotlib, é possível criar desde gráficos de linha e barras simples até visualizações mais sofisticadas, como gráficos de velas (candlestick) e gráficos 3D.

Criada por John D. Hunter em 2003, ela foi desenvolvida para se assemelhar aos recursos de visualização do software MATLAB, permitindo aos usuários gerar gráficos de alta qualidade em uma interface intuitiva.

MATPLOTLIB



Exemplo 1: Gráfico de Linha Simples - Variação de Preço de um ativo ao longo do tempo

```
● ● ●  
import matplotlib.pyplot as plt  
import numpy as np  
  
# Dados de exemplo (Variação diária do preço de uma ação)  
dias = np.arange(1, 11)  
preco_acao = [100, 105, 98, 110, 115, 112, 108, 120, 125, 130]  
  
# Criando o gráfico  
plt.figure(figsize=(8, 5))  
plt.plot(dias, preco_acao, marker='o', linestyle='-',  
color='b')  
  
# Personalizações  
plt.title('Variação do Preço da Ação XYZ')  
plt.xlabel('Dias')  
plt.ylabel('Preço da Ação')  
plt.grid(True)  
  
# Exibindo o gráfico  
plt.show()
```



MATPLOTLIB



Exemplo 2: Gráfico de Linhas Múltiplas - Comparando o Desempenho de Diferentes Ações



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

# Dados de exemplo (Variação diária de preços de diferentes ações)
dias = np.arange(1, 11)
preco_acao1 = [100, 105, 98, 110, 115, 112, 108, 120, 125, 130]
preco_acao2 = [80, 85, 88, 92, 95, 90, 85, 88, 92, 95]
preco_acao3 = [120, 125, 130, 128, 132, 135, 138, 140, 135, 138]

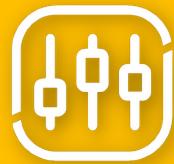
# Criando o gráfico de linhas múltiplas
plt.figure(figsize=(10, 6))
plt.plot(dias, preco_acao1, marker='o', linestyle='--',
         color='b', label='Ação 1')
plt.plot(dias, preco_acao2, marker='s', linestyle='--',
         color='r', label='Ação 2')
plt.plot(dias, preco_acao3, marker='^', linestyle='--',
         color='g', label='Ação 3')

# Personalizações
plt.title('Comparação de Desempenho de Ações')
plt.xlabel('Dias')
plt.ylabel('Preço da Ação')
plt.legend()
plt.grid(True)

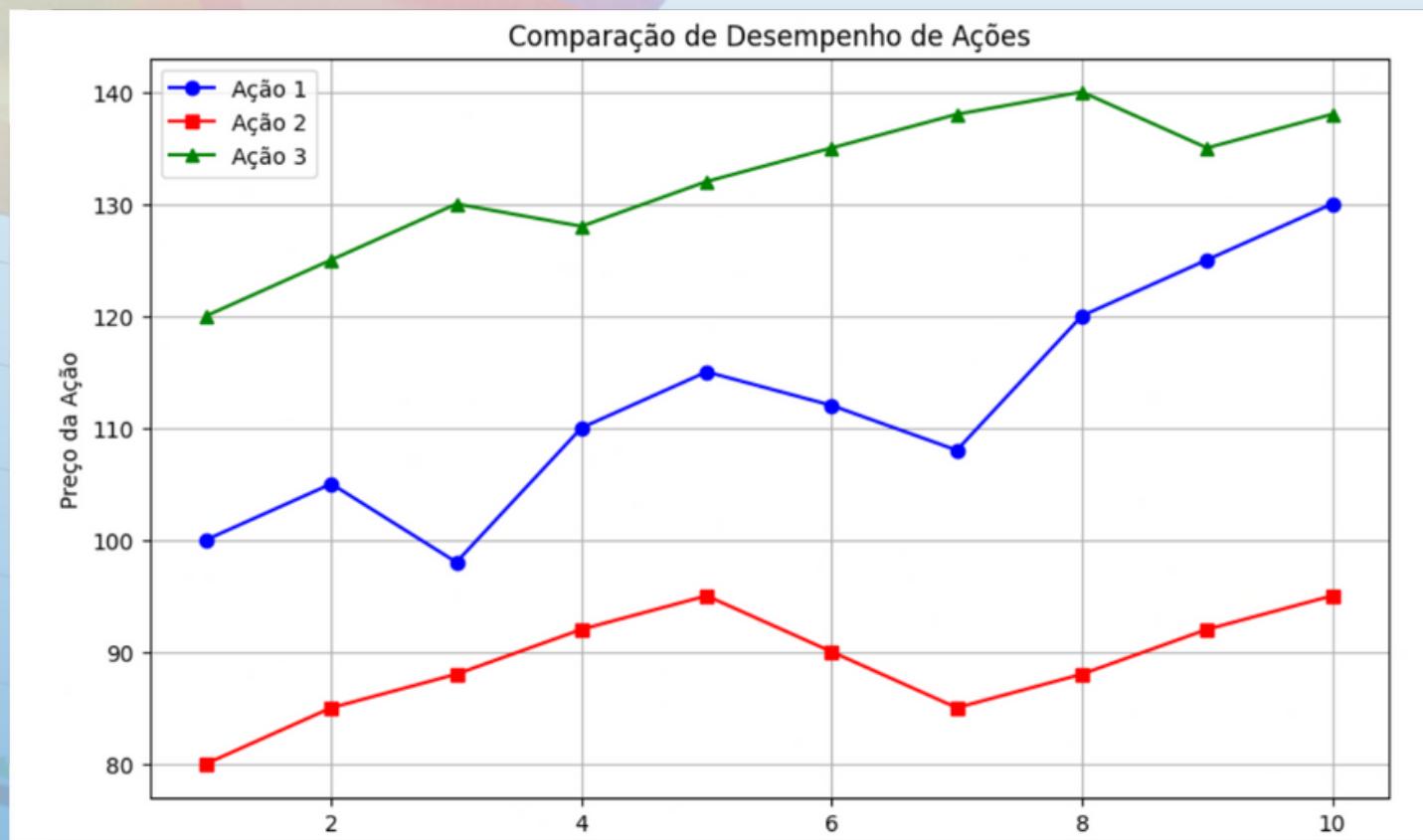
# Exibindo o gráfico
plt.show()
```

Confira o resultado na próxima página

MATPLOTLIB



Exemplo 2: Gráfico de Linhas Múltiplas - Comparando o Desempenho de Diferentes Ações

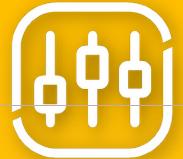


MATPLOTLIB



Exemplo 3: Gráfico de Barras Empilhadas - Alocação de Ativos em um Portfólio de Investimentos

```
● ● ●  
import matplotlib.pyplot as plt  
  
# Dados de exemplo - Alocação de ativos em um portfólio de investimentos  
ativos = ['Ações', 'Títulos', 'Imóveis', 'Caixa']  
portfolio1 = [40, 30, 20, 10] # Alocação em percentual para o Portfolio 1  
portfolio2 = [50, 25, 15, 10] # Alocação em percentual para o Portfolio 2  
  
# Criando o gráfico de barras empilhadas  
plt.figure(figsize=(10, 6))  
plt.bar(ativos, portfolio1, label='Portfolio 1', color='b', width=0.4)  
plt.bar(ativos, portfolio2, bottom=portfolio1, label='Portfolio 2', color='g', width=0.4)  
  
# Personalizações  
plt.title('Alocação de Ativos em Portfólios de Investimentos')  
plt.xlabel('Ativos')  
plt.ylabel('Porcentagem de Alocação')  
plt.legend()  
plt.grid(True)  
  
# Exibindo o gráfico  
plt.show()
```



SEABORN

Seaborn é uma poderosa biblioteca de visualização de dados em Python que é construída em cima do Matplotlib. Ela oferece uma variedade de gráficos estatísticos e ferramentas de personalização, permitindo que os usuários criem visualizações informativas e esteticamente agradáveis. Seaborn é especialmente útil na análise de dados complexos, incluindo dados financeiros, pois facilita a criação de gráficos que destacam relações e padrões importantes nos dados.

No contexto financeiro, o Seaborn pode ser uma ferramenta valiosa para a análise de séries temporais, comparação de ativos, identificação de correlações entre variáveis financeiras e muito mais. Ele complementa o Matplotlib, oferecendo recursos adicionais e estilos de gráficos predefinidos, tornando mais fácil para os profissionais do mercado financeiro criar visualizações impressionantes.

SEABORN



Exemplo 1: Gráfico de Densidade - Distribuição de Retornos de Ações



```
import seaborn as sns
import numpy as np

# Dados de exemplo - Retornos diários de ações
np.random.seed(42)
retornos_acoes1 = np.random.normal(loc=0.001, scale=0.02,
size=1000)
retornos_acoes2 = np.random.normal(loc=0.002, scale=0.015,
size=1000)

# Criando o gráfico de densidade
plt.figure(figsize=(10, 6))
sns.kdeplot(retornos_acoes1, label='Ação 1', shade=True)
sns.kdeplot(retornos_acoes2, label='Ação 2', shade=True)

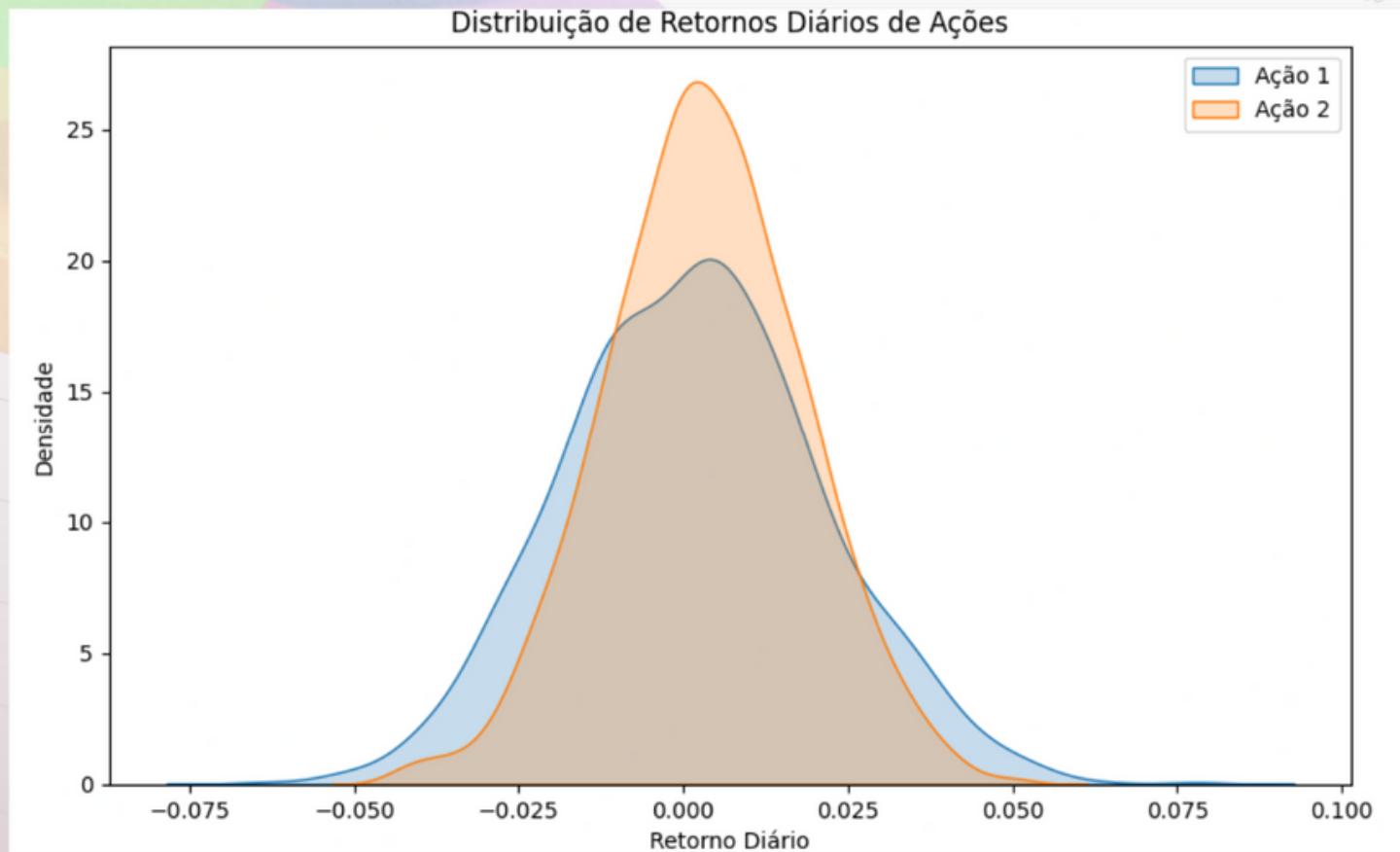
# Personalizações
plt.title('Distribuição de Retornos Diários de Ações')
plt.xlabel('Retorno Diário')
plt.ylabel('Densidade')
plt.legend()

# Exibindo o gráfico
plt.show()
```

SEABORN



Exemplo 1: Gráfico de Densidade - Distribuição de Retornos de Ações



SEABORN



Exemplo 2: Gráfico de Barra - Média de Retornos de Diferentes Ativos



```
import seaborn as sns
import numpy as np

# Dados de exemplo - Média de retornos anuais de diferentes ativos
ativos = ['Ação 1', 'Ação 2', 'Ação 3', 'Ação 4']
media_retornos = [0.1, 0.15, 0.12, 0.08]

# Criando o gráfico de barra
plt.figure(figsize=(8, 6))
sns.barplot(x=ativos, y=media_retornos, palette='pastel')

# Personalizações
plt.title('Média de Retornos Anuais por Ativo')
plt.xlabel('Ativo')
plt.ylabel('Média de Retorno Anual')

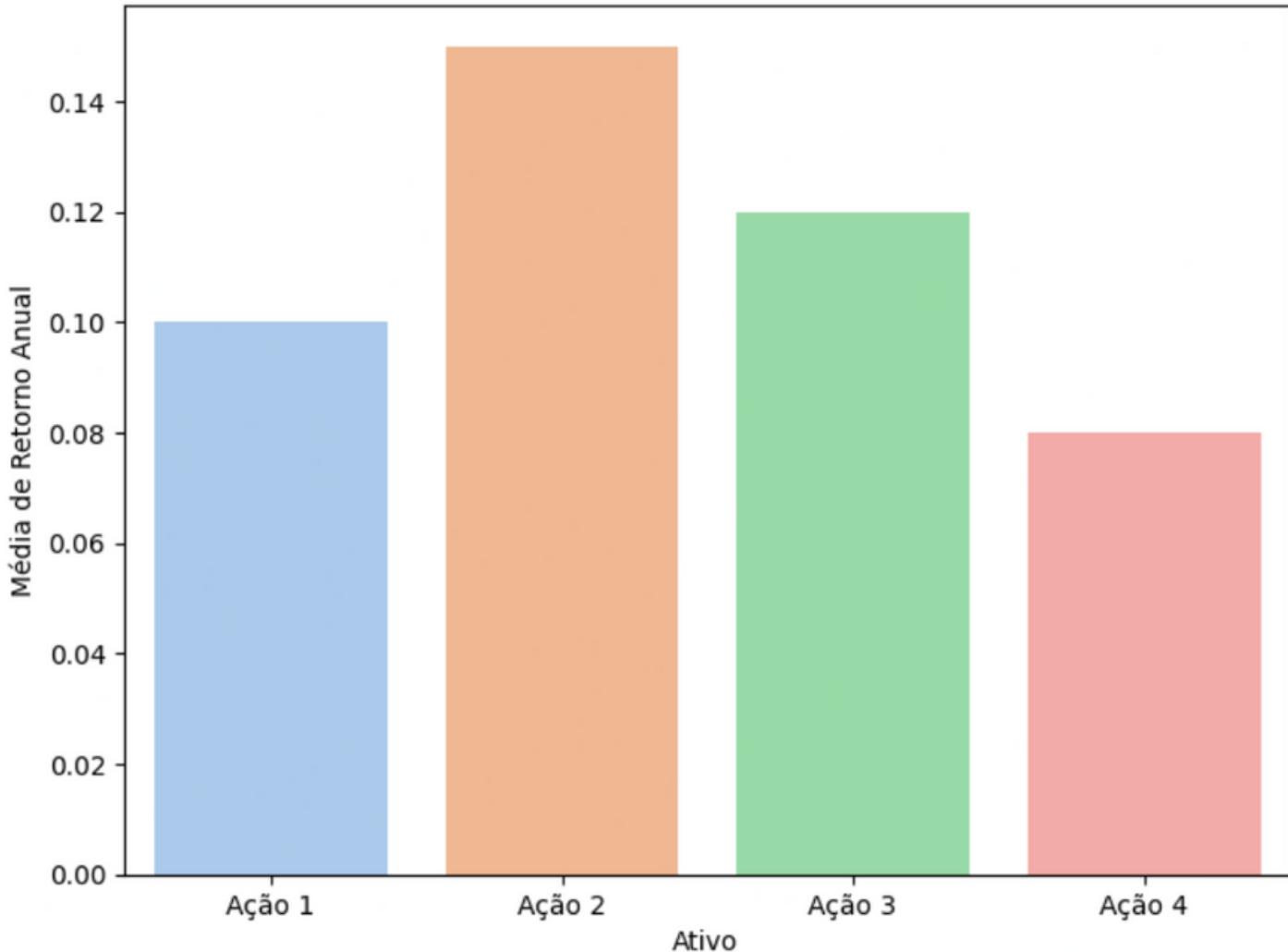
# Exibindo o gráfico
plt.show()
```

SEABORN



Exemplo 2: Gráfico de Barra - Média de Retornos de Diferentes Ativos

Média de Retornos Anuais por Ativo



SEABORN



Exemplo 3: Heatmap - Correlação entre Retornos de Ativos



```
import seaborn as sns
import pandas as pd

# Dados de exemplo - Retornos diários de ações
np.random.seed(42)
data = pd.DataFrame({
    'Ação 1': np.random.normal(loc=0.001, scale=0.02,
size=100),
    'Ação 2': np.random.normal(loc=0.002, scale=0.015,
size=100),
    'Ação 3': np.random.normal(loc=0.003, scale=0.025,
size=100),
})

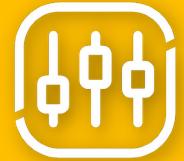
# Calculando a matriz de correlação
correlation_matrix = data.corr()

# Criando o heatmap
plt.figure(figsize=(8, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')

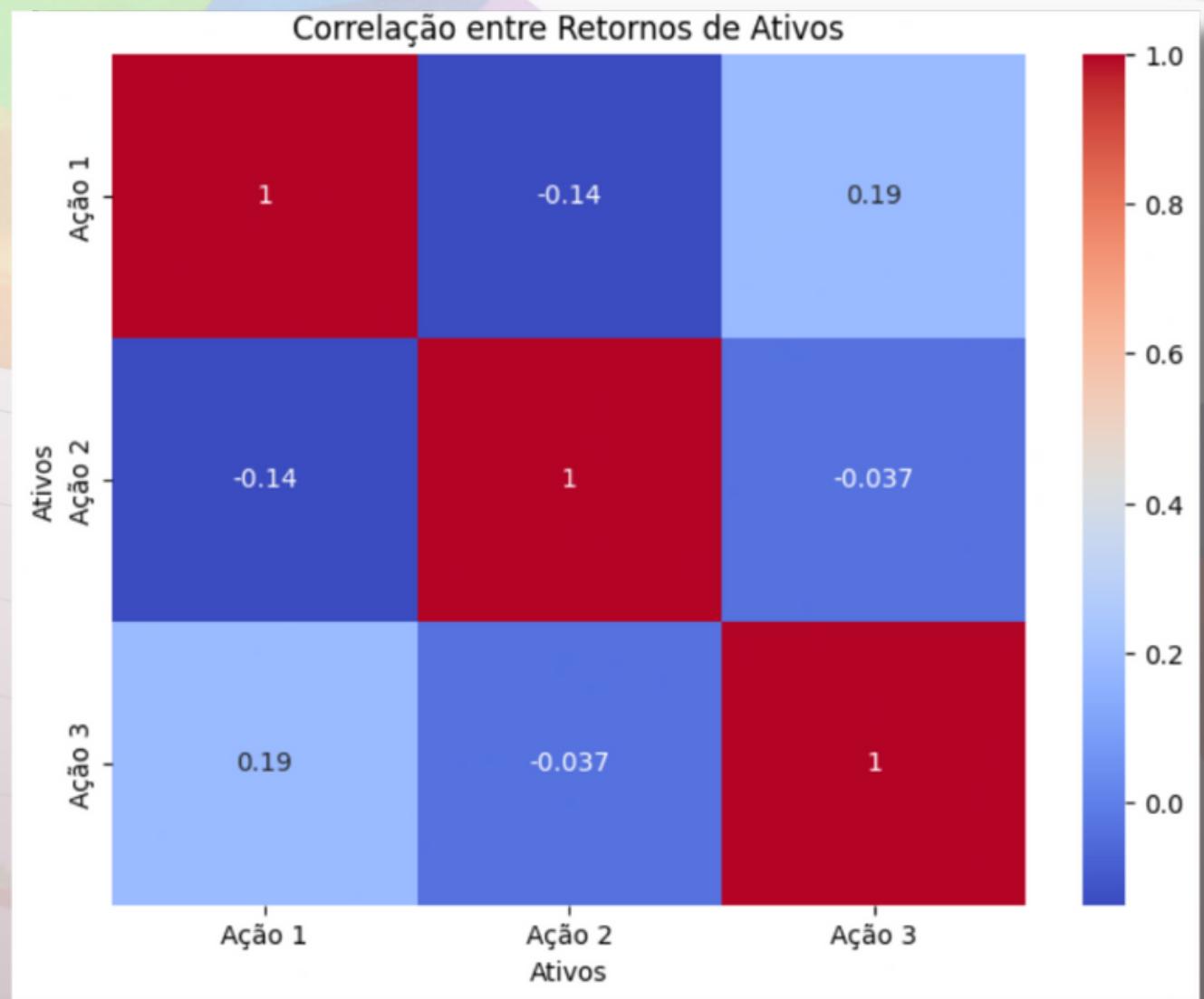
# Personalizações
plt.title('Correlação entre Retornos de Ativos')
plt.xlabel('Ativos')
plt.ylabel('Ativos')

# Exibindo o gráfico
plt.show()
```

SEABORN



Exemplo 3: Heatmap - Correlação entre Retornos de Ativos



PLOTLY



Plotly é uma biblioteca de visualização interativa de dados em Python que permite criar gráficos interativos e altamente personalizáveis. Ela oferece suporte a uma ampla variedade de tipos de gráficos, incluindo gráficos de linha, barras, dispersão, superfícies 3D e muito mais. O Plotly é especialmente adequado para o mercado financeiro, pois permite criar gráficos dinâmicos e interativos que facilitam a análise e exploração de dados complexos.

Além da sua versão em Python, o Plotly também é compatível com outras linguagens de programação, como R, JavaScript e Julia, o que contribui para a sua popularidade em ambientes de análise de dados e ciência de dados.



Exemplo 1: Gráfico de Linha Interativo - Variação de Preços de Ações ao Longo do Tempo

```
● ● ●

import plotly.graph_objects as go
import pandas as pd

# Dados de exemplo (Variação de preços de uma ação ao longo do tempo)
data = pd.date_range('2023-07-01', periods=10, freq='D')
preco_acao = [100, 105, 98, 110, 115, 112, 108, 120, 125, 130]

# Criando o gráfico de linha interativo
fig = go.Figure()
fig.add_trace(go.Scatter(x=data, y=preco_acao,
mode='lines+markers', name='Preço da Ação'))

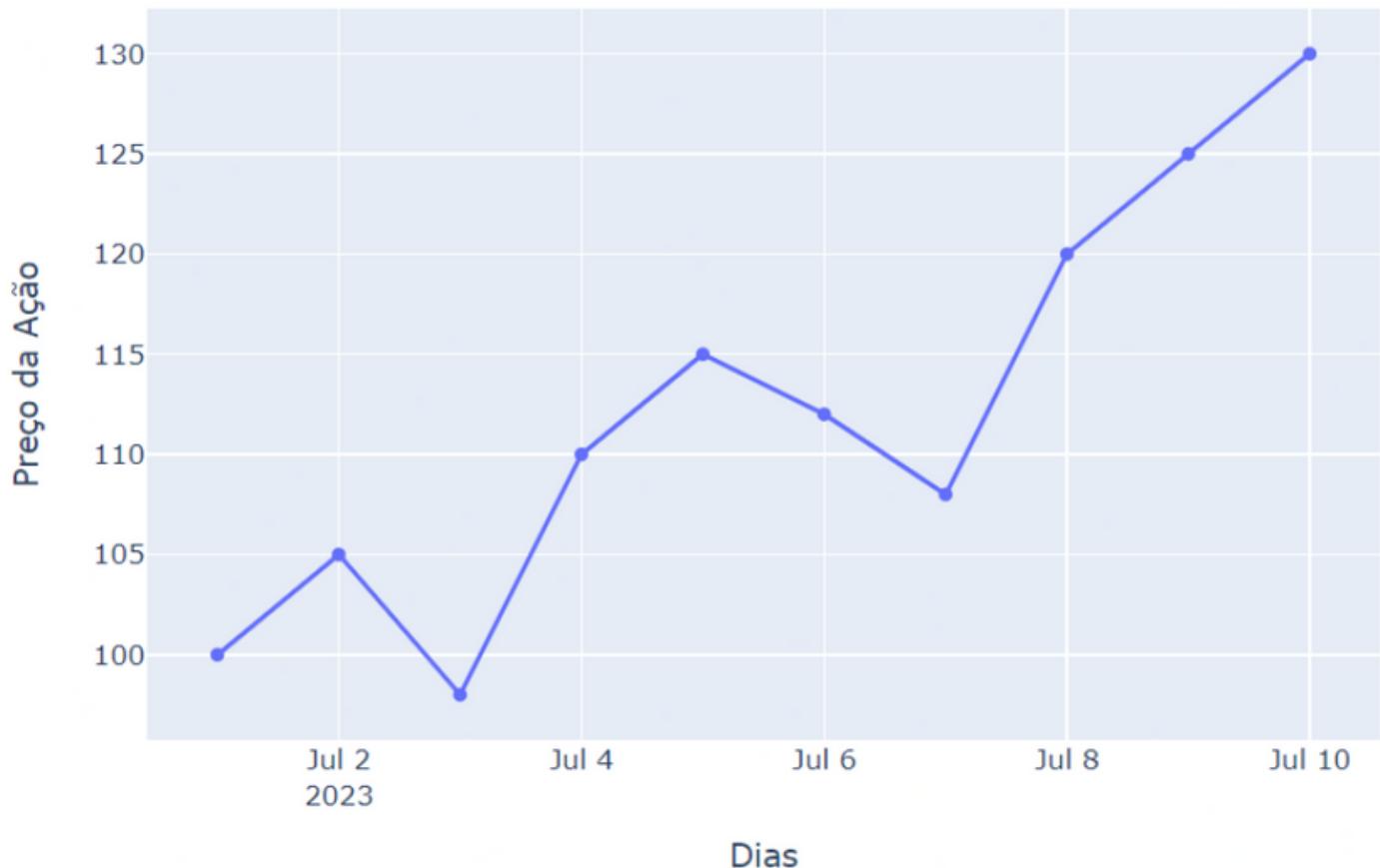
# Personalizações
fig.update_layout(title='Variação do Preço da Ação XYZ',
                  xaxis_title='Dias',
                  yaxis_title='Preço da Ação',
                  width=700,
                  height=500)

# Exibindo o gráfico interativo
fig.show()
```



Exemplo 1: Gráfico de Linha Interativo - Variação de Preços de Ações ao Longo do Tempo

Variação do Preço da Ação XYZ





Exemplo 2: Gráfico de Pizza Interativo - Alocação de Ativos em um Portfólio de Investimentos



```
import plotly.graph_objects as go

# Dados de exemplo - Alocação de ativos em um portfólio de investimentos
ativos = ['Ações', 'Títulos', 'Imóveis', 'Caixa']
porcentagem_alocacao = [40, 30, 20, 10]

# Criando o gráfico de pizza
fig = go.Figure(data=[go.Pie(labels=ativos,
values=porcentagem_alocacao, hole=0.3)])

# Personalizações
fig.update_layout(title='Alocação de Ativos em um Portfólio de Investimentos')

# Exibindo o gráfico interativo
fig.show()
```



Exemplo 2: Gráfico de Pizza Interativo - Alocação de Ativos em um Portfólio de Investimentos





Exemplo 3: Gráfico de Candlestick - Variação de Preços de uma Ação ao Longo do Tempo

```
● ● ●  
import plotly.graph_objects as go  
import yfinance as yf  
  
# Obtendo dados financeiros com yfinance  
ticker = "PETR4.SA" # Símbolo da ação da Apple  
dados_acao = yf.download(ticker, start="2023-07-01",  
end="2023-07-20")  
  
# Criando o gráfico de candlestick  
fig = go.Figure(data=[go.Candlestick(x=dados_acao.index,  
open=dados_acao['Open'],  
high=dados_acao['High'],  
low=dados_acao['Low'],  
close=dados_acao['Close'])])  
  
# Personalizações  
fig.update_layout(title=f'Variação de Preços da Ação  
{ticker}',  
xaxis_title='Data',  
yaxis_title='Preço da Ação',  
xaxis_rangeslider_visible=False,  
width=700,  
height=500)  
  
# Exibindo o gráfico interativo  
fig.show()
```



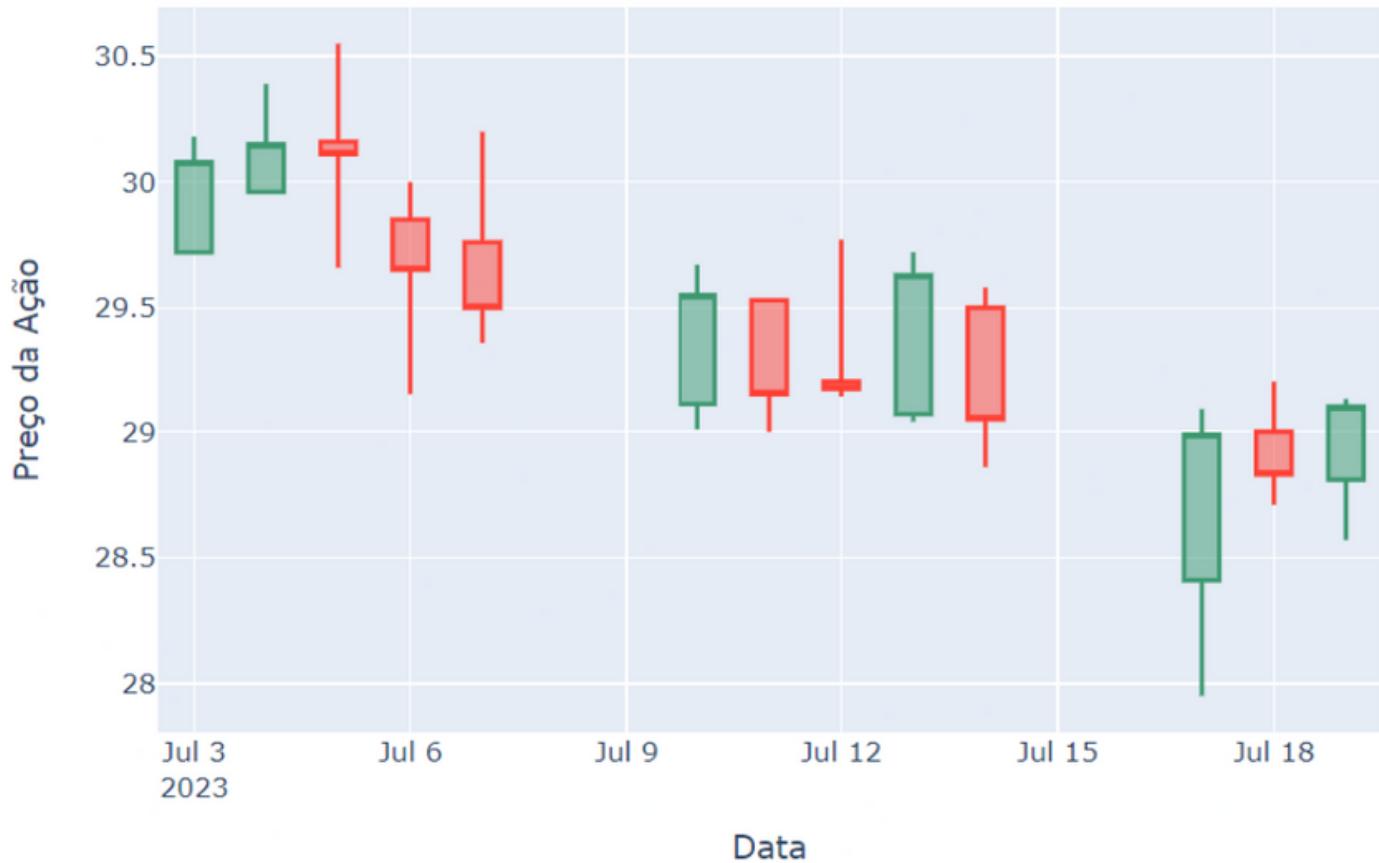
Exemplo 3: Gráfico de Candlestick - Variação de Preços de uma Ação ao Longo do Tempo

```
● ● ●  
import plotly.graph_objects as go  
import yfinance as yf  
  
# Obtendo dados financeiros com yfinance  
ticker = "PETR4.SA" # Símbolo da ação da Apple  
dados_acao = yf.download(ticker, start="2023-07-01",  
end="2023-07-20")  
  
# Criando o gráfico de candlestick  
fig = go.Figure(data=[go.Candlestick(x=dados_acao.index,  
open=dados_acao['Open'],  
high=dados_acao['High'],  
low=dados_acao['Low'],  
close=dados_acao['Close'])])  
  
# Personalizações  
fig.update_layout(title=f'Variação de Preços da Ação  
{ticker}',  
xaxis_title='Data',  
yaxis_title='Preço da Ação',  
xaxis_rangeslider_visible=False,  
width=700,  
height=500)  
  
# Exibindo o gráfico interativo  
fig.show()
```



Exemplo 3: Gráfico de Candlestick - Variação de Preços de uma Ação ao Longo do Tempo

Variação de Preços da Ação PETR4.SA



GOSTOU?



**Conheça a biblioteca da
Trading com Dados com mais
referências para continuar
aprendendo!**

<https://tradingcomdados.com.br/biblioteca>