

Distribuição dos Retornos, Retorno Acumulado e Normalização

5. Distribuição dos Retornos

Fundamentação Teórica

- A análise da distribuição dos retornos pode ser feita através de histogramas, que fornecem uma visualização gráfica da frequência dos diferentes retornos. Além disso, podemos calcular métricas como a curtose e a simetria para entender melhor a forma da distribuição.

Implementação Prática

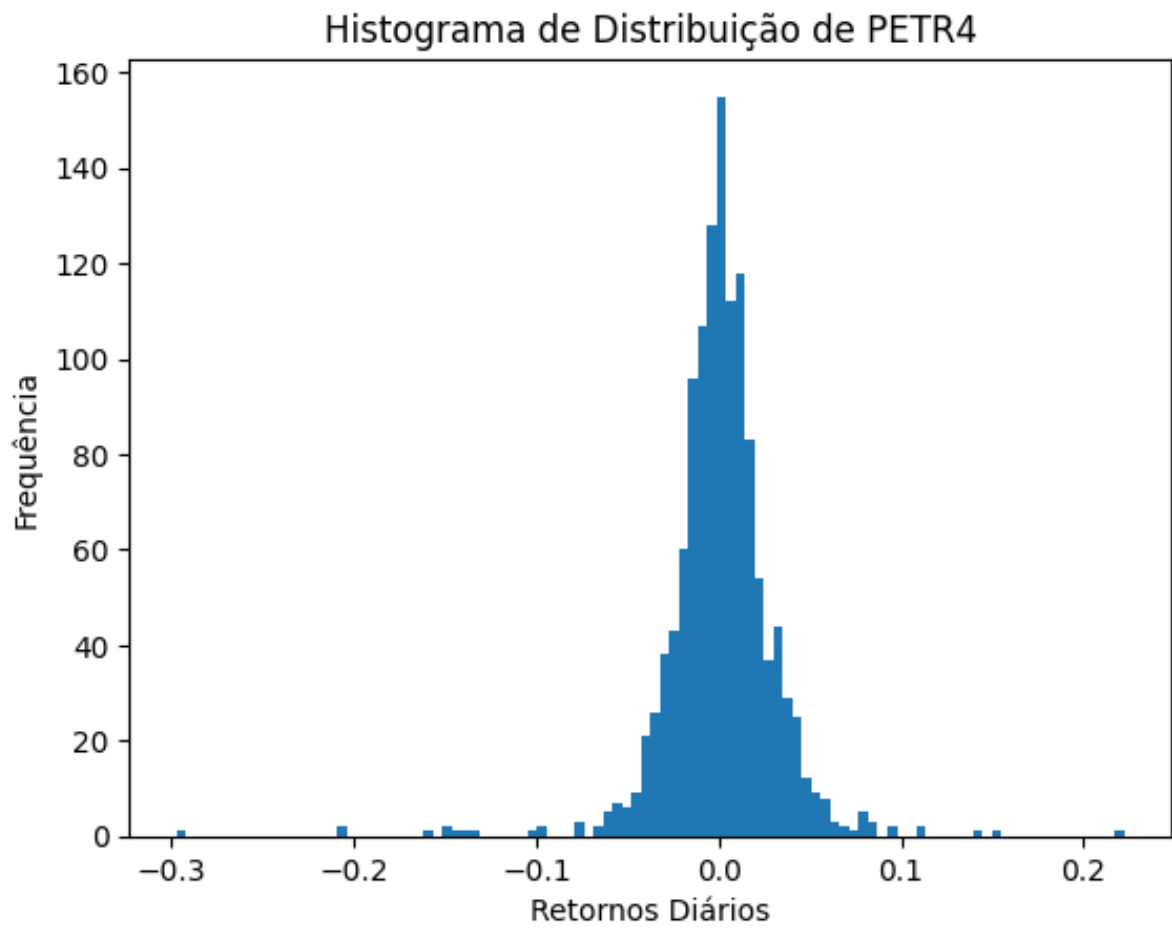
```
In [3]: import yfinance as yf
import matplotlib.pyplot as plt

petro = yf.download('PETR4.SA', start='2017-01-01', end='2022-02-10')

# Calcular os retornos diários
retornos = petro['Adj Close'].pct_change()

# Visualização da distribuição dos retornos
retornos.plot.hist(bins=100, title='Histograma de Distribuição de PETR4')
plt.xlabel('Retornos Diários')
plt.ylabel('Frequência')
plt.show()
```

[*****100%*****] 1 of 1 completed



Conclusão: A distribuição dos retornos é uma ferramenta crucial para entender a frequência e a magnitude dos retornos diários. Analisar essa distribuição ajuda a identificar padrões e a prever possíveis movimentos futuros no mercado.

6. Métricas de Retorno Acumulado

Fundamentação Teórica

- O retorno acumulado é calculado somando os retornos diários e aplicando a fórmula de juros compostos. Isso nos dá uma ideia de quanto um investimento teria crescido (ou diminuído) ao longo do tempo se tivéssemos reinvestido todos os retornos.

Implementação Prática

```
In [8]: petro = yf.download('PETR4.SA', start='2019-01-01', end='2022-05-15')

retornos = petro['Adj Close'].pct_change()

# Calcular o retorno acumulado
ret_acum = (retornos + 1).cumprod()
```

[*****100%*****] 1 of 1 completed

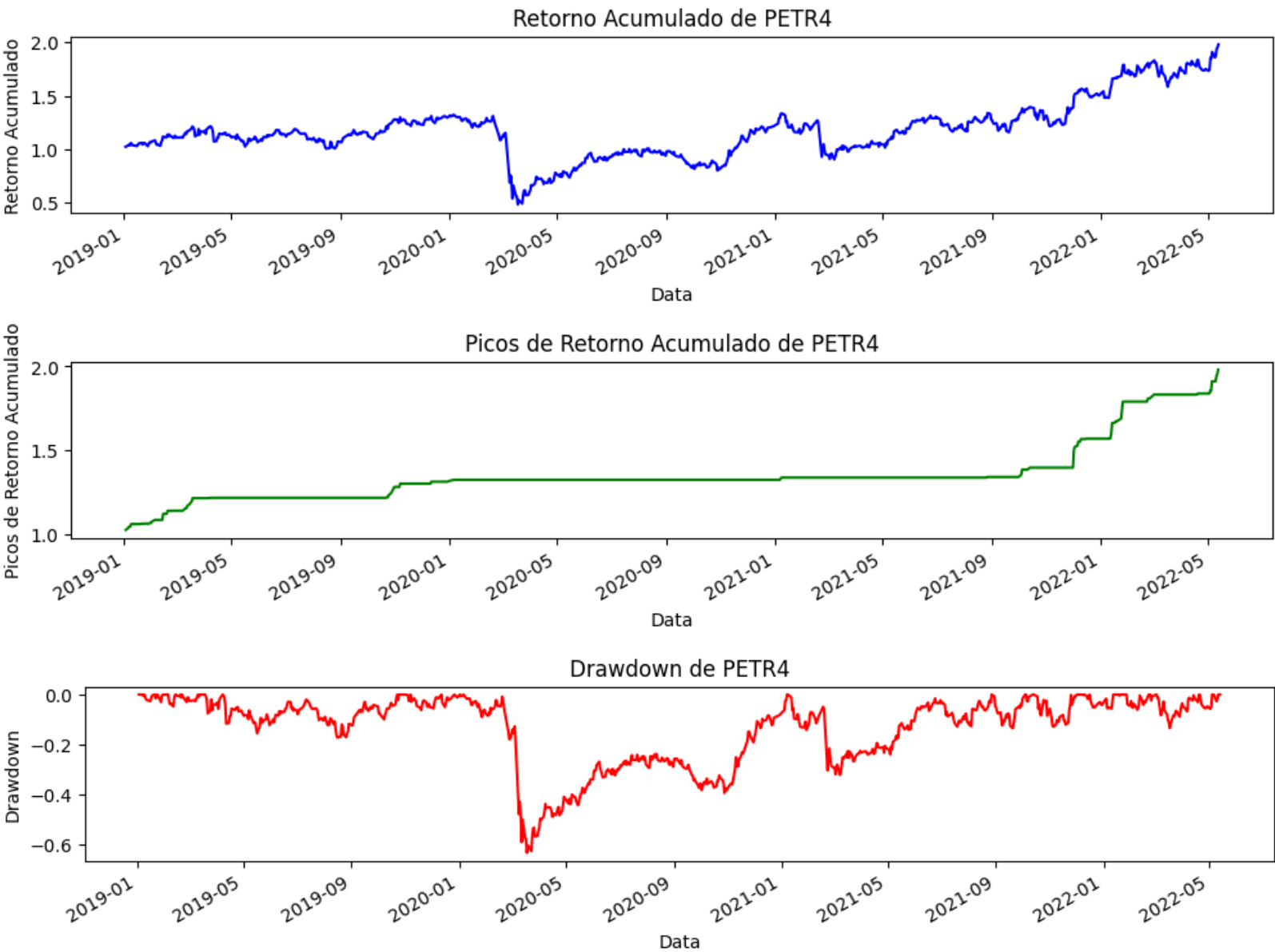
```
In [9]: # Calcular os picos máximos
picos = ret_acum.cummax()

# Calcular o drawdown
drawdown = (ret_acum - picos) / picos
```

```
In [10]: # Plotar o retorno acumulado com a cor azul
ret_acum.plot(figsize=(12, 2), color='blue', title='Retorno Acumulado de PETR4');
plt.xlabel('Data')
plt.ylabel('Retorno Acumulado')
plt.show()

# Plotar os picos máximos com a cor verde
picos.plot(figsize=(12, 2), color='green', title='Picos de Retorno Acumulado de PETR4');
plt.xlabel('Data')
plt.ylabel('Picos de Retorno Acumulado')
plt.show()

# Plotar o drawdown com a cor vermelha
drawdown.plot(figsize=(12, 2), color='red', title='Drawdown de PETR4');
plt.xlabel('Data')
plt.ylabel('Drawdown')
plt.show()
```



Conclusão: As métricas de retorno acumulado são essenciais para avaliar a performance de longo prazo de um ativo. Elas fornecem insights sobre o crescimento potencial e os riscos associados ao investimento, ajudando investidores a tomar decisões mais informadas.

7. Normalização

Fundamentação Teórica

- Normalizamos dados dividindo o valor atual pelo valor inicial, escalando todos os valores para que comecem de uma base comum, geralmente 1. Isso facilita a comparação entre diferentes ativos ao longo do tempo.

Implementação Prática

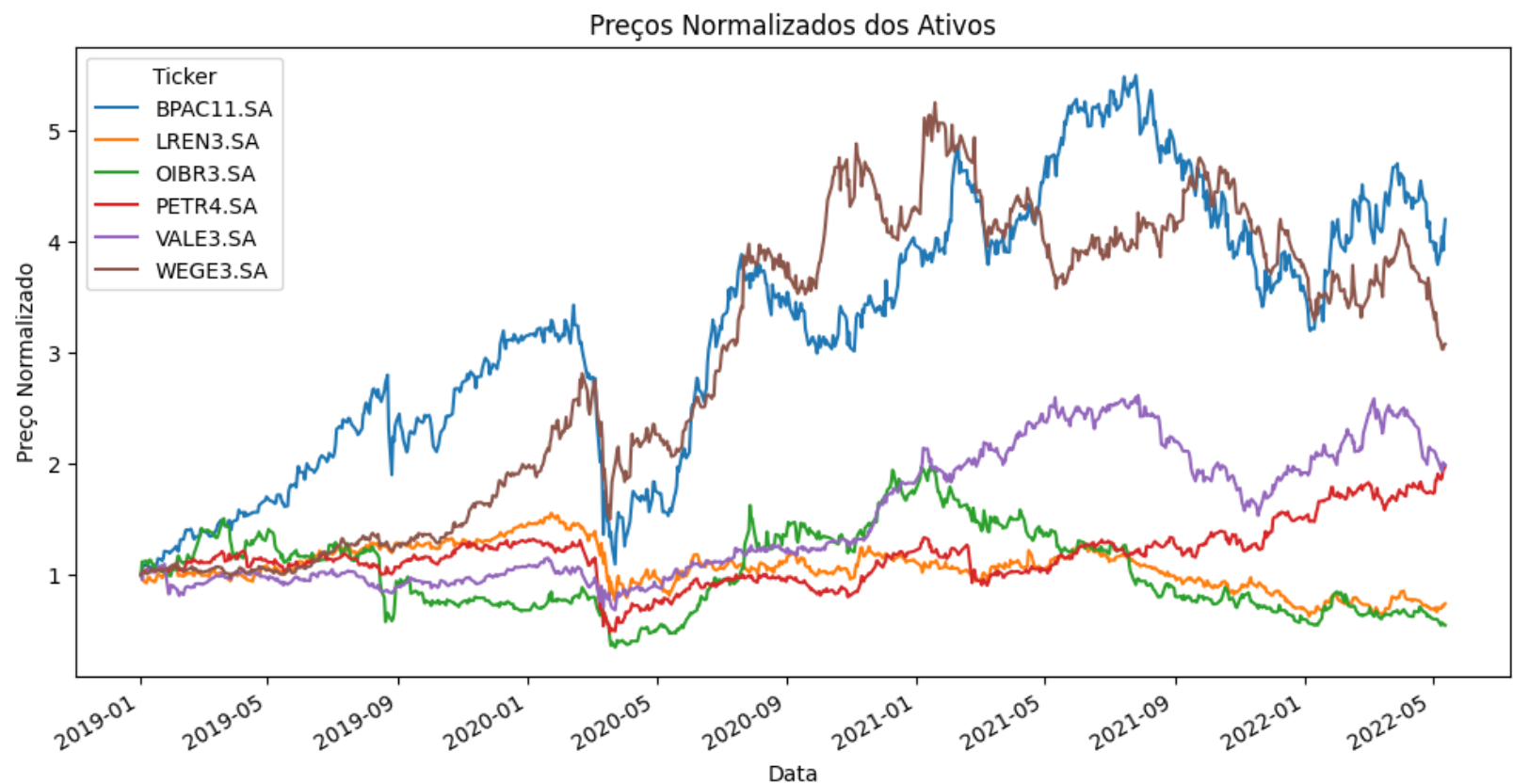
```
In [11]: # Baixar os dados de múltiplos ativos
acoes = ['PETR4.SA', 'WEGE3.SA', 'LREN3.SA', 'VALE3.SA', 'BPAC11.SA', 'OIBR3.SA']
dados = yf.download(acoes, start="2019-01-01", end="2022-05-15")['Adj Close']

# Visualizar os preços ajustados
dados.plot(title='Fechamento dos ativos ao longo do período especificado.', figsize=(12, 6))
plt.xlabel('Data')
plt.ylabel('Preço Ajustado')
plt.show()
```



```
In [12]: # Normalizar os dados na base 1
normalizado = dados / dados.iloc[0]

# Visualizar os dados normalizados
normalizado.plot(title='Preços Normalizados dos Ativos', figsize=(12, 6))
plt.xlabel('Data')
plt.ylabel('Preço Normalizado')
plt.show()
```



Conclusão: A normalização é uma técnica poderosa para comparar a performance de diferentes ativos. Ao ajustar os dados para uma escala comum, podemos identificar quais ativos performaram melhor ou pior ao longo do tempo, auxiliando na análise comparativa.