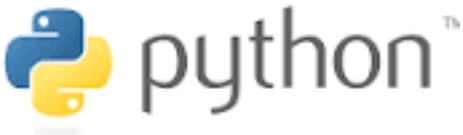
O CÓDIGO DO IMPERADOR

Domine as Finanças com o Poder de Python

PIERRE S. BARBOSA

Introdução

Python é uma das linguagens de programação mais populares no mundo das finanças devido à sua simplicidade, versatilidade e poderosa biblioteca de ferramentas. Este eBook tem como objetivo ensinar conceitos básicos e avançados de finanças utilizando Python, capacitando o leitor a resolver problemas reais do mercado financeiro.



01

Principais Bibliotecas para Finanças

Principais Bibliotecas para Finanças

1. Cálculos Numéricos com NumPy

NumPy é fundamental para realizar cálculos matemáticos e manipular arrays de dados financeiros.

```python
import numpy as np

# Simulando retornos de um investimento
dados = np.array([0.02, 0.03, -0.01, 0.04])

# Cálculo da média de retornos
media = np.mean(dados)
print("Média dos Retornos:", media)

# Cálculo da variância
variancia = np.var(dados)
print("Variância dos Retornos:", variancia)
```

Principais Bibliotecas para Finanças

2. Manipulação de Dados com Pandas

Pandas é essencial para trabalhar com dados estruturados como tabelas e planilhas financeiras.

```
import pandas as pd

# Criando um DataFrame com dados financeiros
dados = {
    'Ano': [2020, 2021, 2022],
    'Lucro': [50000, 60000, 75000]
}

df = pd.DataFrame(dados)
print(df)

# Análise simples: taxa de crescimento
crescimento = df['Lucro'].pct_change()
df['Crescimento'] = crescimento
print(df)
```

Principais Bibliotecas para Finanças

3. Visualização de Dados com Matplotlib e Seaborn

A visualização é essencial para entender tendências financeiras e apresentar resultados.

```
import matplotlib.pyplot as plt

# Lucros anuais
anos = [2020, 2021, 2022]
lucros = [50000, 60000, 75000]

plt.bar(anos, lucros, color='green')
plt.title('Lucros Anuais')
plt.xlabel('Ano')
plt.ylabel('Lucro (R$)')
plt.show()
```

Principais Bibliotecas para Finanças

4. Análise Estatística com Scipy

Scipy é ideal para realizar análises estatísticas, como correlações entre ativos.

```python
from scipy.stats import pearsonr

# Retornos de dois ativos
ativo1 = [0.01, 0.02, -0.01, 0.03]
ativo2 = [0.02, 0.01, 0.00, 0.04]

# Correlação de Pearson
correlacao, _ = pearsonr(ativo1, ativo2)
print("Correlação entre os Ativos:", correlacao)
```

Principais Bibliotecas para Finanças

5. Obtenção de Dados Reais com YFinance

YFinance é uma biblioteca para acessar dados de mercado em tempo real.

```
import yfinance as yf

# Obtendo dados do preço de uma ação
aapl = yf.Ticker("AAPL")
dados = aapl.history(period="1mo")
print(dados[['Open', 'Close']])
```

02

Funções Práticas para Finanças

Funções Práticas para Finanças

1. Calculando Retornos de Investimentos

Retornos ajudam a medir a performance de um ativo financeiro.

```python
import numpy as np

precos = [100, 105, 110, 115]

# Retorno Simples
retornos = [(precos[i] / precos[i-1]) - 1 for i in range(1
print("Retornos Simples:", retornos)

# Retorno Acumulado
retorno_acumulado = np.prod([1 + r for r in retornos]) - 1
print("Retorno Acumulado:", retorno_acumulado)
```

Funções Práticas para Finanças

2. Otimizando um Portfólio

A Teoria de Portfólios ajuda a alocar investimentos de forma eficiente.

```python
import numpy as np

# Dados simulados
pesos = np.array([0.4, 0.6])
retornos = np.array([0.08, 0.12])
cov_matrix = np.array([[0.1, 0.02], [0.02, 0.08]])

# Retorno esperado
retorno_esperado = np.dot(pesos, retornos)
print("Retorno Esperado:", retorno_esperado)

# Risco (desvio padrão)
risco = np.sqrt(np.dot(pesos.T, np.dot(cov_matrix, pesos)))
print("Risco do Portfólio:", risco)
```

Conclusão

Este eBook fornece uma introdução às ferramentas mais importantes do Python para finanças. Pratique os exemplos e explore aplicações para aprimorar suas habilidades no mercado financeiro.

Boa sorte em sua jornada com Python nas finanças!