PANDAS & NUMPY PARA MERCADO FINANCEIRO

@tradingcomdados

Material didático

DataLab

PANDAS & NUMPY PARA MERCADO FINANCEIRO

ESTA APOSTILA TEM COMO OBJETIVO APRESENTAR AS **BIBLIOTECAS NUMPY E PANDAS**, DUAS FERRAMENTAS ESSENCIAIS NO UNIVERSO DA CIÊNCIA DE DADOS E ANÁLISE FINANCEIRA. NUMPY É UMA BIBLIOTECA PODEROSA PARA COMPUTAÇÃO NUMÉRICA EM PYTHON, OFERECENDO RECURSOS PARA MANIPULAÇÃO EFICIENTE DE ARRAYS MULTIDIMENSIONAIS E FUNÇÕES MATEMÁTICAS AVANÇADAS. POR SUA VEZ, O PANDAS É AMPLAMENTE CONHECIDO PELA SUA EFICIÊNCIA EM MANIPULAÇÃO E ANÁLISE DE DADOS ESTRUTURADOS, ESPECIALMENTE NA FORMA DE DATAFRAMES. AMBAS AS BIBLIOTECAS DESEMPENHAM UM **PAPEL FUNDAMENTAL NO MERCADO FINANCEIRO**, PERMITINDO SIMULAÇÕES, CÁLCULOS ESTATÍSTICOS, GERENCIAMENTO DE PORTFÓLIOS E ANÁLISES DE RISCO. AO LONGO DESTA APOSTILA, EXPLORAREMOS EXEMPLOS PRÁTICOS DESSAS APLICAÇÕES, DEMONSTRANDO COMO ESSAS FERRAMENTAS PODEM SER EMPREGADAS PARA TOMAR DECISÕES INFORMADAS E ESTRATÉGICAS NO CONTEXTO FINANCEIRO. SEJA VOCÊ UM INVESTIDOR, ANALISTA OU ENTUSIASTA DA ÁREA FINANCEIRA, AS HABILIDADES APRESENTADAS AQUI SERÃO VALIOSAS PARA POTENCIALIZAR SUAS ANÁLISES E IMPULSIONAR SUAS TOMADAS DE DECISÃO NO MERCADO.

PANDAS & NUMPY PARA MERCADO FINANCEIRO

1. SIMULAÇÃO DE RETORNOS DIÁRIOS DE UMA AÇÃO

NESTE EXEMPLO, UTILIZAREMOS A DISTRIBUIÇÃO NORMAL PARA SIMULAR OS RETORNOS DIÁRIOS DE UMA AÇÃO DURANTE UM PERÍODO DE TEMPO. ESSA SIMULAÇÃO É ÚTIL PARA AVALIAR CENÁRIOS DE RISCO E INCERTEZA NO MERCADO FINANCEIRO.

```python
import numpy as np

# Parâmetros da simulação
media_retorno = 0.001  # Retorno médio diário
desvio_padrao_retorno = 0.02  # Desvio padrão do retorno diário
dias_simulacao = 252  # Número de dias de simulação

# Simulação dos retornos diários
retornos_diarios = np.random.normal(media_retorno,
desvio_padrao_retorno, dias_simulacao)
}
```

PANDAS & NUMPY PARA MERCADO FINANCEIRO

NumPy

2. CÁLCULO DO VALOR PRESENTE LÍQUIDO (VPL)

O VALOR PRESENTE LÍQUIDO É UMA TÉCNICA DE ANÁLISE DE INVESTIMENTOS QUE PERMITE AVALIAR A ATRATIVIDADE DE UM PROJETO OU INVESTIMENTO. NESTE EXEMPLO, UTILIZAREMOS O NUMPY PARA CALCULAR O VPL DE UM FLUXO DE CAIXA.

```
# Fluxo de caixa do investimento
fluxo_caixa = np.array([-1000, 300, 300, 300, 300])

# Taxa de desconto (custo de oportunidade)
taxa_desconto = 0.1

# Cálculo do VPL
vpl = np.npv(taxa_desconto, fluxo_caixa)
print("VPL: {:.2f}".format(vpl))
```

PANDAS & NUMPY PARA MERCADO FINANCEIRO

NumPy

3. RETORNO DE UMA CARTEIRA DE AÇÕES

NESTE EXEMPLO, CALCULAREMOS O RETORNO DE UMA CARTEIRA DE AÇÕES PONDERADA PELOS PESOS DE CADA ATIVO.

```
# Retornos diários de cada ativo da carteira
retorno_ativo1 = np.array([0.01, 0.02, -0.01, 0.005, 0.015])
retorno_ativo2 = np.array([0.015, -0.005, 0.03, 0.01, -0.02])

# Pesos de cada ativo na carteira
peso_ativo1 = 0.6
peso_ativo2 = 0.4

# Cálculo do retorno da carteira
retorno_carteira = (retorno_ativo1 * peso_ativo1) +
(retorno_ativo2 * peso_ativo2)
print("Retorno da Carteira: {}".format(retorno_carteira))
```

PANDAS & NUMPY PARA MERCADO FINANCEIRO

NumPy

4. VOLATILIDADE HISTÓRICA DE UMA AÇÃO

A VOLATILIDADE HISTÓRICA MEDE A VARIAÇÃO DOS PREÇOS DE UMA AÇÃO AO LONGO DO TEMPO. NESTE EXEMPLO, CALCULAREMOS A VOLATILIDADE HISTÓRICA DE UMA AÇÃO UTILIZANDO NUMPY.

```
# Preços diários da ação
precos_acao = np.array([100, 102, 98, 105, 99])

# Retornos diários
retornos_acao = precos_acao[1:] / precos_acao[:-1] - 1

# Volatilidade histórica
volatilidade = np.std(retornos_acao) * np.sqrt(252)  # 252 é o
número de dias úteis em um ano
print("Volatilidade Histórica: {:.2f}%".format(volatilidade *
100))
```

PANDAS & NUMPY PARA MERCADO FINANCEIRO

5. GERAÇÃO DE NÚMEROS ALEATÓRIOS PARA SIMULAÇÃO DE CENÁRIOS

NESTE EXEMPLO, UTILIZAREMOS O NUMPY PARA GERAR NÚMEROS ALEATÓRIOS QUE PODEM REPRESENTAR SIMULAÇÕES DE CENÁRIOS, COMO TAXAS DE JUROS OU VARIAÇÕES DE PREÇOS.

```python
# Simulação de 10 cenários de taxas de juros
taxas_juros = np.random.uniform(0.05, 0.08, 10)
print("Taxas de Juros Simuladas:")
print(taxas_juros)

# Simulação de 5 cenários de variações percentuais
variacao_precos = np.random.normal(0.02, 0.01, 5)
print("Variações Percentuais Simuladas:")
print(variacao_precos)
```

PANDAS & NUMPY PARA MERCADO FINANCEIRO

pandas

1. CRIAÇÃO E MANIPULAÇÃO DE UM DATAFRAME DE DADOS FINANCEIROS

NESTE EXEMPLO, CRIAREMOS UM DATAFRAME NO PANDAS PARA ARMAZENAR DADOS FINANCEIROS, COMO PREÇOS DE AÇÕES, VOLUMES DE NEGOCIAÇÃO E DATAS.

```python
import pandas as pd

# Dados financeiros de exemplo
data = pd.date_range(start='2023-01-01', periods=5, freq='D')
precos_acao = [100, 102, 98, 105, 99]
volumes_negociacao = [1000, 800, 1200, 1500, 900]

# Criação do DataFrame
df_dados_financeiros = pd.DataFrame({'Data': data, 'Preço':
precos_acao, 'Volume': volumes_negociacao})
print(df_dados_financeiros)
```

PANDAS & NUMPY PARA MERCADO FINANCEIRO

pandas

2. LEITURA DE DADOS FINANCEIROS DE UM ARQUIVO CSV

NESTE EXEMPLO, FAREMOS A LEITURA DE UM ARQUIVO CSV CONTENDO DADOS FINANCEIROS DE UMA AÇÃO E UTILIZAREMOS O PANDAS PARA MANIPULAR OS DADOS.

```
# Leitura dos dados do arquivo CSV
df_dados_financeiros = pd.read_csv('dados_financeiros.csv')

# Exibição dos 5 primeiros registros
print(df_dados_financeiros.head())
```

PANDAS & NUMPY PARA MERCADO FINANCEIRO

pandas

3. CÁLCULO DE MÉDIA MÓVEL SIMPLES (SMA)

A MÉDIA MÓVEL SIMPLES É UMA TÉCNICA UTILIZADA NO MERCADO FINANCEIRO PARA SUAVIZAR A SÉRIE DE PREÇOS DE UM ATIVO E IDENTIFICAR TENDÊNCIAS. NESTE EXEMPLO, UTILIZAREMOS O PANDAS PARA CALCULAR A SMA DE UMA AÇÃO.

```
# Dados financeiros de exemplo
data = pd.date_range(start='2023-01-01', periods=10, freq='D')
precos_acao = [100, 102, 98, 105, 99, 101, 97, 103, 104, 100]

# Criação do DataFrame
df_dados_financeiros = pd.DataFrame({'Data': data, 'Preço':
precos_acao})

# Cálculo da Média Móvel Simples com janela de 3 dias
df_dados_financeiros['SMA'] =
df_dados_financeiros['Preço'].rolling(window=3).mean()
print(df_dados_financeiros)
```

PANDAS & NUMPY PARA MERCADO FINANCEIRO

pandas

4. ANÁLISE DE CORRELAÇÃO ENTRE AÇÕES

A ANÁLISE DE CORRELAÇÃO É ÚTIL PARA ENTENDER COMO O RETORNO DE UMA AÇÃO ESTÁ RELACIONADO AO RETORNO DE OUTRA AÇÃO. NESTE EXEMPLO, CALCULAREMOS A CORRELAÇÃO ENTRE DUAS AÇÕES USANDO O PANDAS.

```
# Dados de retorno diário de duas ações
retorno_acao1 = [0.01, 0.02, -0.01, 0.005, 0.015]
retorno_acao2 = [0.015, -0.005, 0.03, 0.01, -0.02]

# Criação do DataFrame
df_retornos = pd.DataFrame({'Ação1': retorno_acao1, 'Ação2':
retorno_acao2})

# Cálculo da correlação entre as ações
correlacao = df_retornos['Ação1'].corr(df_retornos['Ação2'])
print("Correlação entre as Ações: {:.2f}".format(correlacao))
```

PANDAS & NUMPY PARA MERCADO FINANCEIRO

pandas

5. CONCATENAÇÃO DE DADOS DE DIFERENTES ATIVOS

A CONCATENAÇÃO DE DADOS É ÚTIL PARA UNIR INFORMAÇÕES DE DIFERENTES ATIVOS EM UMA ÚNICA ESTRUTURA DE DADOS. NESTE EXEMPLO, UTILIZAREMOS O PANDAS PARA CONCATENAR DADOS DE DUAS AÇÕES.

```
# Dados de uma ação
data_acao1 = pd.date_range(start='2023-01-01', periods=5,
freq='D')
precos_acao1 = [100, 102, 98, 105, 99]

# Dados de outra ação
data_acao2 = pd.date_range(start='2023-01-01', periods=5,
freq='D')
precos_acao2 = [50, 52, 48, 55, 49]

# Criação dos DataFrames
df_acao1 = pd.DataFrame({'Data': data_acao1, 'Preço':
precos_acao1})
df_acao2 = pd.DataFrame({'Data': data_acao2, 'Preço':
precos_acao2})

# Concatenação dos DataFrames
df_concatenado = pd.concat([df_acao1, df_acao2])
print(df_concatenado)
```