

USD/JPY 分析 ダッシュボード

リアルタイム分析と
バックテストによるト
レード最適化
中島麻莉- 2024.12.19

プロジェクト概要

目的:

- 為替データを利用したリアルタイム分析と自動取引履歴管理
- 簡単にトレードシグナルと結果を可視化

使用技術:

- プログラミング言語: Python ver3.12.4
- ライブラリ: Dash, Matplotlib, Pandas, yfinance, SQLite ver 3.26.0

システム構成

データ取得:

- Yahoo Finance API ('yfinance'ライブラリ) を用いた為替データの取得

分析手法:

- 移動平均線 (SMA)
- RSI (相対力指数)
- 移動平均クロスオーバー戦略

記録管理:

- SQLite データベースを用いた取引履歴の保存と総利益の計算

主な機能

リアルタイムデータ取得:

- USD/JPY の為替レートデータ取得

テクニカル分析:

- 移動平均線、RSIの計算
- 売買シグナル生成

ダッシュボード:

- 売買ボタンと直近の取引履歴の表示
- グラフによる可視化 (価格推移、RSIチャート)

取引履歴:

- 最新の3件の取引を表示

ダッシュボードの構成

価格推移チャート:

- 売買シグナルをマーキング (緑: Buy, 赤: Sell)

RSIチャート:

- オーバーボート (>70) とオーバーソールド (<30) を視覚化

直近の取引履歴:

- 各取引の日時、アクション、価格、利益を表示

売買ボタン:

- ボタン押下で取引を記録し、利益を更新

テクニカル分析手法

移動平均線 (SMA):

- 短期 (10日) と長期 (50日) のクロスオーバーを用いて売買シグナル生成

RSI (14日):

- 買われすぎ (>70) と売られすぎ (<30) の判定

バックテスト結果

バックテスト:

- クロスオーバー戦略を適用して総利益を計算

結果:

- 総利益: [変数`total_return` の値]

SQLITE データベース

取引記録:

- 日時、アクション (Buy/Sell)、価格、利益を記録

最新の3件:

- 表形式で表示

実行手順

データ取得:

- ``fetch_fx_data`` 関数で取得

分析:

- ``calculate_sma``, ``calculate_rsi`` でテクニカル指標を計算
- ``generate_signals`` でシグナル生成

バックテスト:

- ``backtest_strategy`` で総利益を計算

ダッシュボード起動:

- ``run_dashboard`` でインタラクティブな可視化

今後の展望

改良点:

- 複数通貨ペアの対応
- ボラティリティ分析や他のテクニカル指標の追加

応用例:

- 自動売買システムへの拡張
- 学習モデルによる予測精度の向上