

Ksim V2.1 - 智慧型多資產交易回測系統 (AI-Assisted Trading Simulator)

Ksim V2.1 是一個基於 Python 與 Streamlit 構建的互動式金融回測引擎。本專案不僅是一個交易模擬器，更是一個展示軟體架構重構 (Refactoring) 與 AI 輔助開發 (AI-Assisted Development) 的實作案例

(連readme也是AI輸出的，如果有相關回饋可以加我DC)。

 線上體驗 Demo: [Ksim V2.1 App](#)  Discord: [點擊加我好友](#)

開發故事：從 V2 到 V2.1 的架構演進

專案背景

Ksim 的初衷是建立一個能驗證交易邏輯的無風險沙盒。

AI 協作開發模式 (Human-AI Collaboration)

本專案採用「人類設計邏輯，AI 實作架構」的開發模式：

1. **Design (我)**: 定義金融交易規則(如保證金計算、強制平倉邏輯)、介面需求與系統變數設計(如觀測期與可視範圍分離)。
2. **Implementation (Gemini)**: 協助撰寫 Python 程式碼、偵測邏輯漏洞(Debug)、優化數據處理效率，並協助進行模組化拆分。

架構重構 (Refactoring)

- **V2 版本 (Legacy)**:
 - 初期僅由 app.py (介面) 與 data_manager.py (資料) 組成。
 - 缺點: 隨著功能增加(如槓桿、圖表)，程式碼過於臃腫，邏輯與 UI 耦合度高，難以維護。
- **V2.1 版本 (Current)**:
 - 為了提升系統的可維護性與擴充性，我將系統拆分為 5 個獨立模組。
 - 採用類似 MVC (Model-View-Controller) 的設計思維，讓資料流更清晰。

系統模組結構

以下是 V2.1 優化後的檔案結構與職責說明：

檔案名稱	模組類型	職責說明
app.py	Frontend / Controller	主程式入口。負責 UI 渲染、處理使用者互動(按鈕/輸入)，並調用後端邏輯。
logic.py	Backend Logic	核心交易引擎。處理資金計算、訂單撮合、保證金維持率檢測、強制平倉(Liquidation)邏輯。
charts.py	Visualization	視覺化模組。專注於 Plotly 圖表

檔案名稱	模組類型	職責說明
		繪製，包含 K 線、MA 均線、動態價格標籤與指標顯示。
data_manager.py	Data / ETL	資料處理層。負責串接 Yahoo Finance API，進行資料清洗 (Cleaning) 與技術指標 (RSI, MA) 計算。
config.py	Configuration	全域配置檔。集中管理所有參數 (手續費率、槓桿限制、MA 週期)，方便一次性調整系統規則。

✨ 核心功能介紹

1. 多資產支援 (Multi-Asset Support)

一套系統同時兼容三種截然不同的金融商品，自動切換最小交易單位與代碼檢查規則：

-  股票 (Stock)：如 TSLA, NVDA (單位：股)
-  外匯 (Forex)：如 JPY=X, EURUSD=X (單位：點)
-  加密貨幣 (Crypto)：如 BTC-USD, ETH-USD (單位：顆)

2. 進階模擬機制 (Advanced Simulation)

- 觀測期與視野分離：
 - 初始觀測期 (**Observation**)：模擬開始前預跑 250 天，確保長天期均線 (MA120) 數值正確。
 - 圖表可視範圍 (**View**)：介面僅聚焦最近 100 天的走勢，避免 K 線被壓縮，提供最佳視覺體驗。
- 保證金交易 (**Margin Trading**)：
 - 支援 1x ~ 20x 槓桿 (**Leverage**) 設定。
 - 內建做空 (**Shorting**) 機制，可模擬熊市獲利策略。
 - 風險控管：即時計算維持保證金，觸發條件自動執行 強制平倉 (**Liquidation**)。

3. 視覺化互動圖表

- 使用 **Plotly** 繪製專業級互動圖表。
- 解決了傳統圖表標籤對齊問題，動態顯示持倉成本線、止損 (SL) 與止盈 (TP) 線。

🔧 本地安裝與執行 (Local Installation)

如果你想在自己的電腦上運行此專案：

1. 環境需求

- Python 3.10 或以上 (建議 3.13)

2. 安裝依賴套件

```
pip install streamlit pandas numpy yfinance plotly
```

3. 執行程式 在專案目錄下執行：

```
streamlit run app.py
```