**Required skills**

Experience in python programing in stock trading

**References**

**MUST** read about Tradingview.com

**MUST** read about Alpaca.market, API Trading

**MUST** read about https://handsoffinvesting.com/calculate-and-analyze-rsi-using-python/

**Scope**

To develop a program preferably in python, to trade stocks based on its Relative Strength Index (RSI).

**What is RSI**

The Relative Strength Index (**RSI**), is a momentum oscillator that measures the speed and change of price movements. The **RSI** oscillates between zero and 100. Traditionally the **RSI** is considered overbought when above 70 and oversold when below 30.

**High level summary**

**RSI** is a momentum oscillator commonly used to predict when a stock is oversold or overbought. This indicator is not always accurately predict stop price, or not all stocks are accurately predictable by **RSI**.

**Goals and Deliverables:**

1. Goal 1, is to screen stocks to identify which stocks can be predictable with >50% accuracy using **RSI** as indicator. This goal is actual replication of a work presented in <https://handsoffinvesting.com/calculate-and-analyze-rsi-using-python/>. The top 20 hits will be called RSIDS-20 (RSI dependent stocks 20).
2. Goal 2, is to develop a program to trade **RSDIS** utilizing a strategy illustrated in the figure 1 and to back test the developed program on **RSIDS-20** to identify RSIDS-5 tradable stocks.
3. Goal 3 is to set up the program to run the program and execute RSIDS-5 trades in Alpaca trading platform.

Figure 1: Illustration of RSI trading execution based on RSI strategy. The RSIDS are bought at several stages when RSI is decreasing towards 0. The RSIDS are sold at several stages when RSI is increasing towards 100.



Hey There,

I have just add your strategy to my bot using your mention RSI method in Crypto currency pairs. Also we could gather any data form Alpaca api. Please find attachment sample materials for BTC/USD price attached below.

Thank you

I working on trading bot with following futures:

-> With custom Strategy: Witten in python, using pandas. Example MACD crossover, EMA crossover, RSI, and all the technical indicators are included

-> Downloads market data: Download historical data of the exchange and the markets you may want to trade.

-> Backtest and Test strategy on downloaded historical data.

-> Optimize: Find the best parameters for your strategy using hyper optimization which employs machining learning methods.

-> Optimize buy, sell, take profit (ROI), stop-loss, and trailing stop-loss parameters for your strategy.

-> Select markets or pairs: Create your static list of pairs or use an automatic one based on the top traded volumes and/or prices (not

available during backtesting). You can also explicitly blacklist markets you don't want to trade.

-> Run: Test your strategy with simulated money (Dry-Run mode) or deploy it with real money (Live-Trade mode).

- >Run using Edge (optional module): The concept is to find the best historical trade expectancy by markets based on a variation of the stop-loss and then allow/reject markets to trade. The sizing of the trade is based on the risk of a percentage of your capital.

-> Control/Monitor: Use Te-legram or a REST API tdameritrade (start/stop the bot, show profit/loss, daily summary, current open trades results, etc.).

-> Analyse: Further analysis can be performed on either Backtesting data or trading history (SQL database), including automated standard plots, and methods to load the data into interactive environments.