Kronos-Backtester Documentation

Overview

"Kronos-Backtester" is a Python-based toolkit for financial data analysis and backtesting of trading strategies.

It provides an effective platform for simulating trading strategies using historical data to evaluate their performance

and potential profitability.

Installation Guide

pip install kronos-backtester

This command installs "Kronos-Backtester" along with its necessary dependencies.

Requirements and Dependencies

- Python 3.x
- Pandas
- yfinance (for fetching financial data)
- Additional Python libraries as needed for specific strategies.

Code Documentation

1. Function: momentum_trading_strategy (Example Strategy)

- Description: This function demonstrates a momentum trading strategy. It is an example of how a strategy function

should be structured for use with "Kronos-Backtester." The function calculates moving averages over specified

windows to generate trading signals.

- Parameters:
  - data: DataFrame with 'Date', 'Price', and additional columns for analysis.
  - short_window: Short-term moving average window.
  - long_window: Long-term moving average window.
  - entry_threshold: Minimum price change to trigger a buy/sell signal (percentage).
  - exit_threshold: Minimum price change to exit a position (percentage).
- Returns: DataFrame with buy/sell signals. The function must return:
  - 1 for a buy signal.
  - -1 for a sell signal.
  - 0 to hold (no action).
- Note: This strategy serves as a template. Users can develop their own strategies ensuring they return the specified
  signal values.

2. Class: Backtester

- Method: __init__
  - Description: Initializes the Backtester class with a given strategy function.
  - Parameters:
    - strategy: A callable strategy function to be tested.
- Method: testTickerReport
  - Description: Tests the given strategy on a stock ticker over a period and generates a performance report.
  - Parameters: Ticker symbol, start date, end date.
  - Returns: A report detailing the performance of the strategy, including metrics like net worth, equity final,
    max drawdown, and Sharpe Ratio.

3. Function: wrapper

- Description: A wrapper function for the strategy function, setting default parameters for ease of use.

- Parameters: The wrapper function should accept the same parameters as the strategy function.

- Usage Example:

```
bt = Backtester(wrapper)
test = bt.testTickerReport('AAPL', '2010-01-01', '2020-01-01')
for key in test:
    print(key, test[key])
```

Usage and Examples

In this section, detailed examples of setting up and using "Kronos-Backtester" will be provided, including:

- How to initialize the Backtester with a strategy.
- Executing a backtest on historical stock data.
- Analyzing the output and understanding the performance metrics.

Troubleshooting and FAQs

Common questions and issues will be addressed here, along with guidance on how to resolve typical problems
encountered during the setup and use of "Kronos-Backtester."

Contribution Guidelines

Guidelines for contributing to the "Kronos-Backtester" project, including coding standards, pull request procedures,
and contact information for the project maintainers.

License and Credits

Details of the project's license, acknowledgments to contributors, and any third-party resources or libraries used in
the development of "Kronos-Backtester."