

Торговля коинтегрированными парами



Савинов Владислав

Теоретические источники

Pairs Trading: Performance of a Relative Value Arbitrage Rule, 1998 by E. Gatev



В основе работы лежит стратегия торговли коинтегрированными парами, разработанная в середине 1980х годов.

Statistical Arbitrage: High Frequency Pairs Trading, 2014 by Ruben Joakim Gundersen



Стратегия предполагает сопоставление длинной и короткой позиций по двум коинтегрированным активам.

High Frequency and Dynamic Pairs Trading Based on Statistical Arbitrage Using a Two-Stage Correlation and Cointegration Approach, 2014 by George J. Miao



Практические источники

Pairs trading: Python template & strategy review by Mikayil Majidov



Торговая стратегия для торговли
коинтегрированными парами
акций



В своей работе я также опирался на ряд источников, предоставляющих варианты технической реализации теоретической модели

Описание стратегии

1

Выбор множества пар активов

Критерий выбора: коинтегрированность цен активов

2

Анализ показателей

Первый этап наблюдений: построение сигналов, вычисление коэффициента Шарпа и отношения максимальной доходности к максимальной просадке на тестовом временном интервале. Цель: выбрать пару, которой будем торговать в будущем

3

Торговля

Второй этап наблюдений: запуск стратегии на контрольном интервале

1 Выбор множества пар активов

- Входные данные: тикеры доступных для торговли активов и их цены закрытия



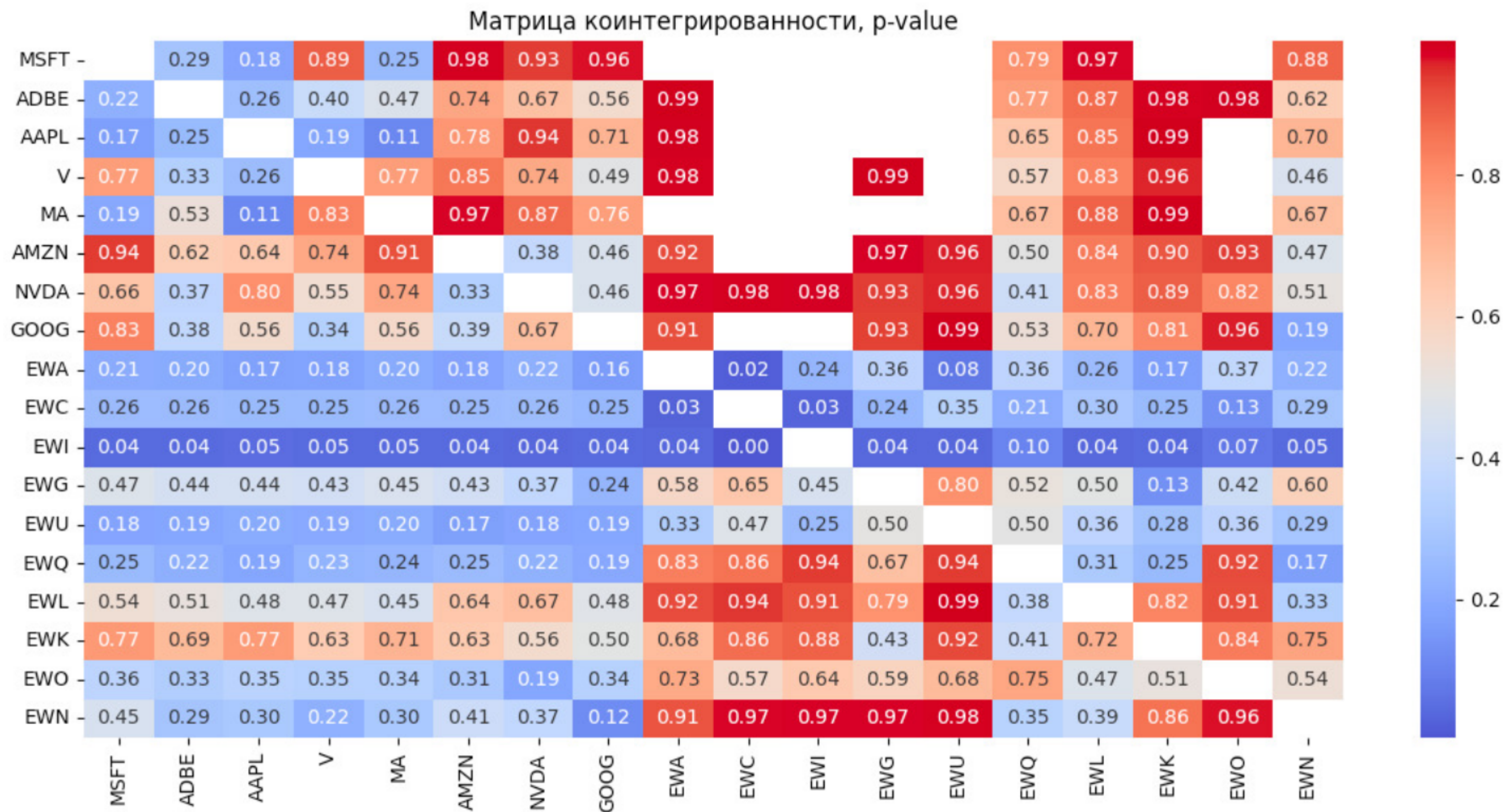
Скачивание данных из yahoo finance по ценам осуществляется при помощи библиотеки `pandas_datareader`

- Среди этих данных находим все коинтегрированные активы на тестовом временном промежутке — от '2010-01-01' до '2019-12-31'



Для проверки коинтеграции применена использующая тест Энгла-Грэнджера функция `coint` из библиотеки `statsmodels.tsa.stattools`

- Коэффициент значимости α берем равным 0.05

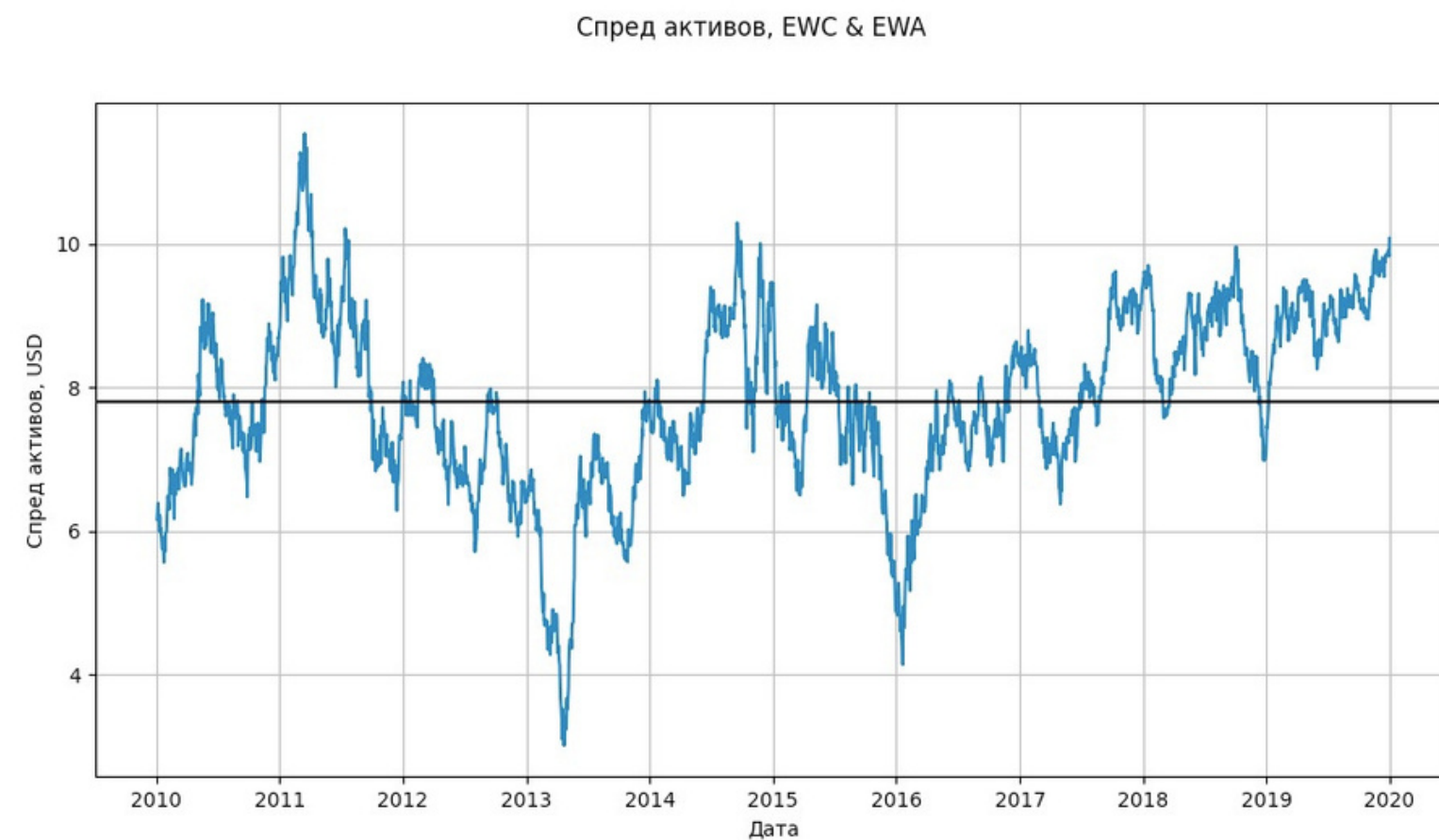


Матрица коинтегрированности тестируемых активов для наглядности выводится пользователю в формате heatmap

Цены закрытия по дням, EWC & EWA



Также выводится график изменения цен по указанным парам активов



$\text{price_A} - \text{price_B} * \text{beta}$, где beta — параметр линейной регрессии

```
import statsmodels.api as sm

xs = sm.add_constant(pricesB)
model = sm.OLS(pricesA, xs)
result = model.fit()
hedgeRatio = result.params[1]
spread = pricesA - pricesB * hedgeRatio
```

Вычисляется спред пар активов и выводится график, чтобы пользователь мог убедиться в стационарности

2 Анализ показателей

Берется спред активов в выбранных парах . Когда значение пересекает z-score, совершается сделка. При этом если было преодолено значение z-score-high, осуществляется покупка первого актива в лонг позицию, второго — в шорт; если же преодолено значение z-score-low — наоборот. При пересечениях получаются сигналы для покупки

```
signals = pd.DataFrame()
signals[self._tickerA] = self._tickerAData
signals[self._tickerB] = self._tickerBData
spread, hedge = self._worker.getSpread()
signals['z-low'], signals['z-score'], signals['z-up'] = self.getZScore(
    spread)
signals['signalsA'] = 0
signals['signalsA'] = np.select(
    [signals['z-score'] > signals['z-up'],
     signals['z-score'] < signals['z-low']],
    [-1, 1],
    default=0)
signals['signalsB'] = -hedge * signals['signalsA']
```

```

portfolio = pd.DataFrame()
portfolio[self._tickerA] = signals[self._tickerA]
portfolio['holdingsA'] = signals['positionsA'].cumsum() * \
    signals[self._tickerA] * positionsA
portfolio['cashA'] = self._capital / 2 - \
    (signals['positionsA'] *
     signals[self._tickerA] *
     positionsA).cumsum()
portfolio['totalA'] = portfolio['holdingsA'] + portfolio['cashA']
portfolio['returnA'] = portfolio['totalA'].pct_change()
portfolio['positionsA'] = signals['positionsA']

```

```

portfolio['total'] = portfolio['totalA'] + portfolio['totalB']
finalPortfolio = portfolio['total'].iloc[-1]
portfolio = portfolio.dropna()
hDrawdown = max(portfolio['total']) / min(portfolio['total'])
delta = (portfolio.index[-1] - portfolio.index[0]).days
returns = (finalPortfolio /
           self._capital) ** (DAYS_ONE_YEAR / delta) - 1

```

```

def getSharpeRatio(self, returns, days, basePct=0.02):
    """
    Выдает коэффициент Шарпа, получая на вход дневную доходность по отношению к прошлому дню и количество прошедших дней

    Считает безрисковую процентную ставку по дефолту за 2% годовых
    """
    if basePct < 0 or basePct > 1:
        return None
    ret = (np.cumprod(1 + returns) - 1)[-1]
    return (ret - basePct) / (np.sqrt(days) * np.std(1 + returns))

```

Далее происходит вычисление возвратов по каждому активу и получение статистики по торговле на тестовом временном интервале. Согласно этой статистике отбираем ту пару, у которой сумма коэффициентов Шарпа наибольшая, либо незначительно отличается от максимума, но при этом отношение наибольшей доходности к максимальной просадке максимально

3 Торговля

Цены закрытия по дням, EWA & EWC -- выбрали эту пару для торговли



Результаты торговли

Стартовый капитал: 10000

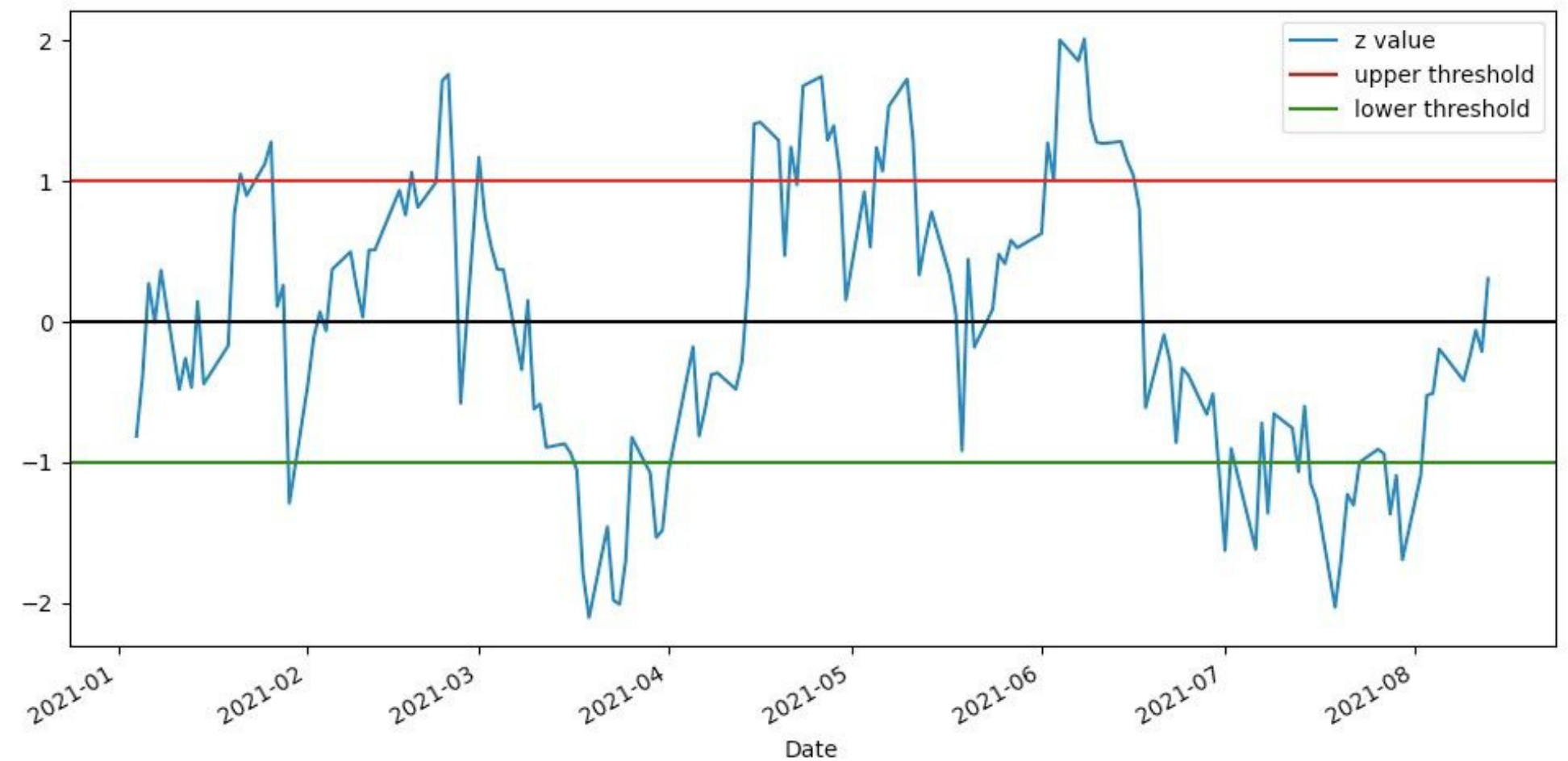
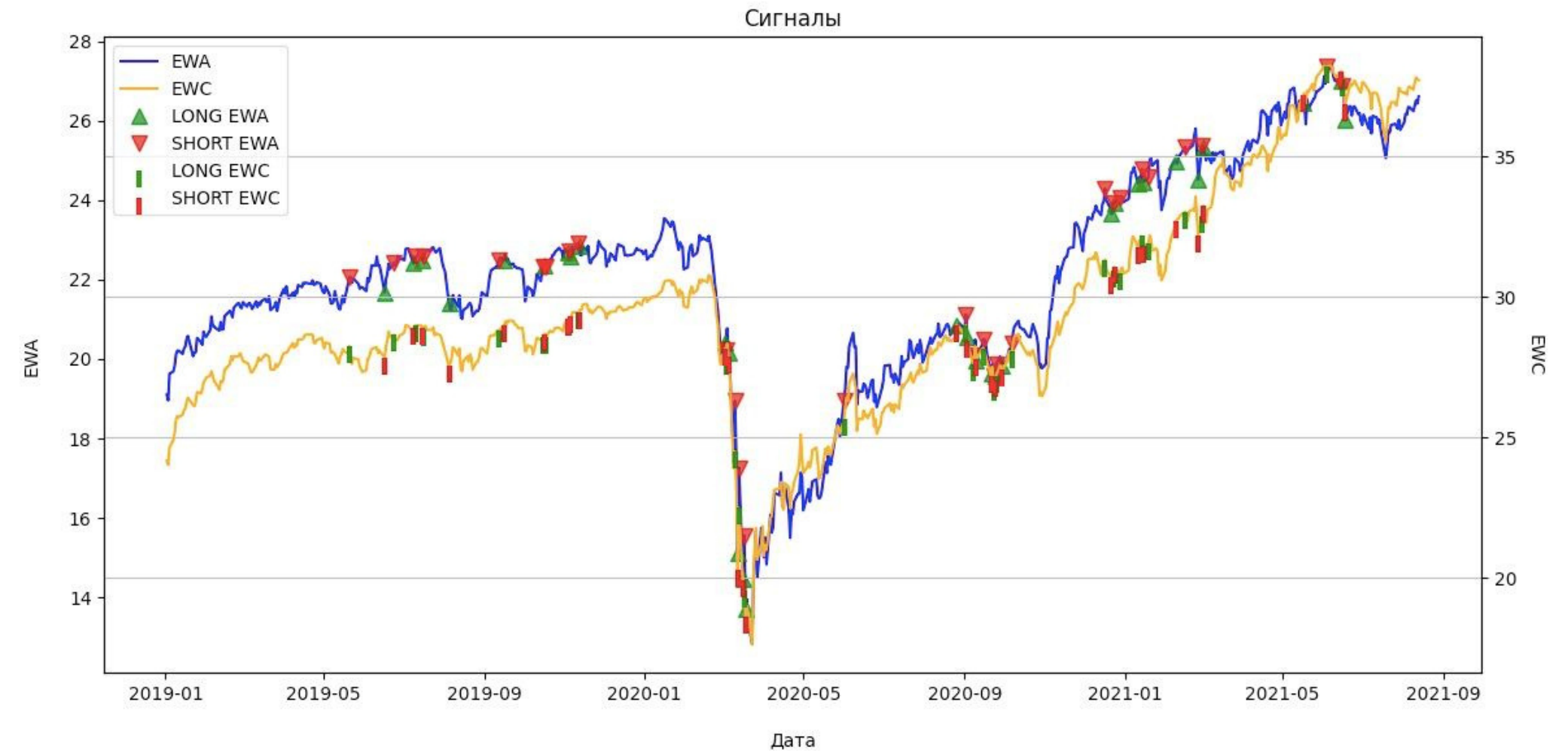
Конечный капитал: 10847.49

Возврат на капитал: 14.52%

Коэффициент Шарпа первого актива: 2.17

Коэффициент Шарпа второго актива: -1.72

Отношение доходности к просадке: 1.08



Наблюдения и преимущества реализации

- В дефолтный список активов в коде были добавлены **ETF** несмотря на то, что в оригинальных статьях торговля происходит на акциях
- Возможность указать путь до файла со списком активов, среди которых искать коинтегрированную пару, через аргументы командой строки

Торговля на ETF показала себя лучше на тестовом периоде — особенно хорошо показывает себя торговля парой Канада + Австралия

```
python3 src/main.py -f input.txt
```


Недостатки модели и пути улучшения результатов

— Коинтегрированность может не сохраняться на контрольном временном интервале

— Определяемая алгоритмом сумма для покупки конкретного актива может не быть кратной стоимости единицы актива — необходимо убедиться, что брокер позволяет покупать дробные части

— Модель не учитывает комиссии брокера, что может существенно отражаться на доходности при HFT



Торговля более чем одной парой активов, например, тремя парами, показавшими наилучшие результаты на тестовой выборке



Введение функции от коэффициента Шарпа, просадки и др. для поиска оптимального распределения капитала между парами при помощи регрессии



Кроме OLS можно использовать orthogonal regression и минимизировать тем самым перпендикулярное расстояние, а не вертикальное



Увеличение частотности данных для тестирования