Soccer Club Stock Arbitrage

Algorithmic Trading Strategy with Full Code

Haixiang

2023.10 | Vol 43.

hxyan.2015@gmail.com | github.com/hxyan2020

STRATEGY & ECONOMIC RATIONALE

The investment universe consists of liquid soccer clubs' stocks that are publicly traded. The investor then sells short stocks of clubs that play UEFA Championship matches (or other important matches) at the end of the business day before the match. Stocks are held for one day, and the portfolio of stocks is equally weighted (if there are multiple clubs with matches that day).

BUY	SELL
The opposite	clubs that play UEFA Championship matches (or other important matches) at the end of the business day before the match.

PARAMETER & VARIABLES

PARAMETER	VALUE
MARKETS TRADED	Equity
FINANCIAL INSTRUMENTS	Stocks
REGION	Europe
PERIOD OF REBALANCING	Daily
NO. OF TRADED INSTRUMENTS	20
WEIGHTING	Equal weighting
LOOKBACK PERIODS	N/A
LONG/SHORT	Long & Short

ALGORITHM

```
class SoccerClubsStocksArbitrage(QCAlgorithm):

    def Initialize(self):
        self.SetStartDate(2000, 1, 1)
        self.SetCash(100000)

        self.tickers = [
            'FCPP',  # Futebol Clube Do Porto
            'SPSO',  # Sporting Clube De Portugal
            'SLBEN', # Benfica
            'LAZI',  # Lazio
            'ASR',   # AS Rome
            'AJAX',  # AJAX
            'JUVE',  # Juventus
            'MANU',  # Manchester United
            'BVB',   # Dortmund
            'CCP',   # Celtic
            # 'BOLA' # Bali Bintang Sejahtera Tbk PT
        ]

        self.match_dates = {}
```

```
        for ticker in self.tickers:
            security = self.AddData(QuantpediaSoccer, ticker, Resolution.Daily)
            security.SetFeeModel(CustomFeeModel())
            security.SetLeverage(5)

        csv_string_file =
self.Download('data.quantpedia.com/backtesting_data/equity/soccer/soccer_matches.csv')
        lines = csv_string_file.split('\r\n')
        for line in lines:
            line_split = line.split(';')
            date = datetime.strptime(line_split[0], "%d.%m.%Y").date()

            self.match_dates[date] = []
            for i in range(1, len(line_split)):
                ticker = line_split[i]
                self.match_dates[date].append(ticker)

    def OnData(self, data):
        self.Liquidate()

        short = []

        # Looking for todays date, because only daily closes are traded.
        today = (self.Time - timedelta(days=1)).date()

        if today in self.match_dates:
            for ticker in self.tickers:
                if ticker in self.match_dates[today] and ticker in data:
                    short.append(ticker)

        for ticker in short:
            self.SetHoldings(ticker, -1 / len(short))

# Quantpedia data.
# NOTE: IMPORTANT: Data order must be ascending (datewise)
class QuantpediaSoccer(PythonData):
    def GetSource(self, config, date, isLiveMode):
        return
SubscriptionDataSource("data.quantpedia.com/backtesting_data/equity/soccer/{0}.csv".format
(config.Symbol.Value), SubscriptionTransportMedium.RemoteFile, FileFormat.Csv)

    def Reader(self, config, line, date, isLiveMode):
        data = QuantpediaSoccer()
        data.Symbol = config.Symbol

        if not line[0].isdigit(): return None
        split = line.split(';')

        data.Time = datetime.strptime(split[0], "%d.%m.%Y") + timedelta(days=1)
        data['price'] = float(split[1])
        data.Value = float(split[1])
```

```
        return data

# Custom fee model.
class CustomFeeModel(FeeModel):
    def GetOrderFee(self, parameters):
        fee = parameters.Security.Price * parameters.Order.AbsoluteQuantity * 0.00005
        return OrderFee(CashAmount(fee, "USD"))
```

BACKTESTING PERFORMANCE

Fig 1. Overall Performance

PSR	0.143%	Sharpe Ratio	0.5
Total Trades	1388	Average Win	2.00%
Average Loss	-1.21%	Compounding Annual Return	8.633%
Drawdown	47.800%	Expectancy	0.269
Net Profit	587.312%	Loss Rate	52%
Win Rate	48%	Profit-Loss Ratio	1.66
Alpha	0.07	Beta	-0.018
Annual Standard Deviation	0.138	Annual Variance	0.019
Information Ratio	0.052	Tracking Error	0.215
Treynor Ratio	-3.779	Total Fees	$14232.19
Estimated Strategy Capacity	$0	Lowest Capacity Asset	JUVE.QuantpediaSoccer 2S
Portfolio Turnover	9.37%		

Fig 2. Performance Metrics

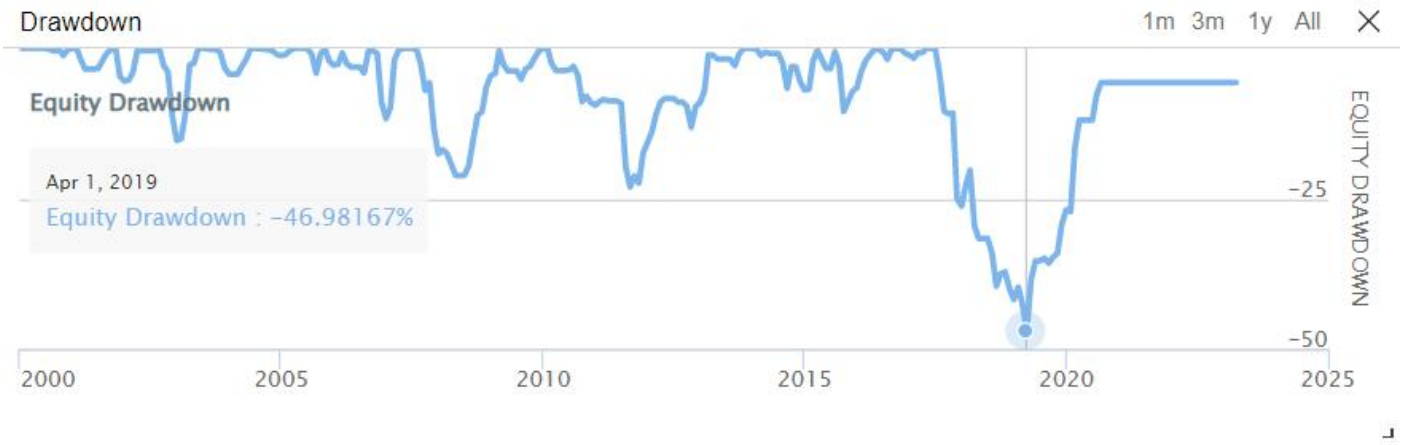

Fig 3. Drawdown

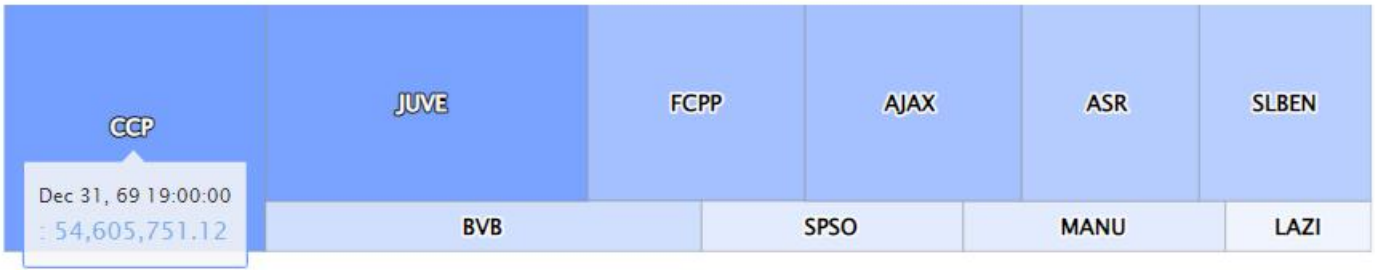

Fig 4. Assets Sales Volume