Not Over Thinking

January Barometer

Algorithmic Trading Strategy with Full Code

Haixiang

2023.10 | Vol 44.

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

STRATEGY & ECONOMIC RATIONALE

Invest in the equity market in each January. Stay invested in equity markets (via ETF, fund, or futures) only if January return is positive; otherwise, switch investments to T-Bills.

BUY	SELL
When Jan return is positive	The opposite

PARAMETER & VARIABLES

PARAMETER	VALUE
MARKETS TRADED	Equity
FINANCIAL INSTRUMENTS	CFDs, ETFs, funds, futures
REGION	Global
PERIOD OF REBALANCING	Monthly
NO. OF TRADED INSTRUMENTS	1
WEIGHTING	Equal weighting
LOOKBACK PERIODS	Month
LONG/SHORT	Long Only

ALGORITHM

```
from AlgorithmImports import *

class JanuaryBarometer(QCAlgorithm):

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

        leverage:int = 5
        data:Equity = self.AddEquity("SPY", Resolution.Daily)
        data.SetLeverage(leverage)
        self.market:Symbol = data.Symbol

        data:Equity = self.AddEquity("SHY", Resolution.Daily)
        data.SetLeverage(leverage)
        self.bond:Symbol = data.Symbol

        self.max_missing_days:int = 5

        self.start_price:float|None = None
        self.recent_month:int = -1

    def OnData(self, data):
        if self.recent_month == self.Time.month:
            return
        self.recent_month = self.Time.month
```

```
        if self.Securities[self.market].GetLastData() and
self.Securities[self.bond].GetLastData():
            if (self.Time.date() -
self.Securities[self.market].GetLastData().Time.date()).days < 5 and (self.Time.date() -
self.Securities[self.bond].GetLastData().Time.date()).days < self.max_missing_days:
                if self.Time.month == 1:
                    self.Liquidate(self.bond)
                    self.SetHoldings(self.market, 1)

                    self.start_price = self.Securities[self.market].Price

                if self.Time.month == 2 and self.start_price:
                    perf:float = self.Securities[self.market].Price / self.start_price - 1
                    if perf > 0:
                        self.SetHoldings(self.market, 1)
                    else:
                        self.start_price = None
                        self.Liquidate(self.market)
                        self.SetHoldings(self.bond, 1)
            else:
                self.Liquidate()
        else:
            self.Liquidate()
```

BACKTESTING PERFORMANCE

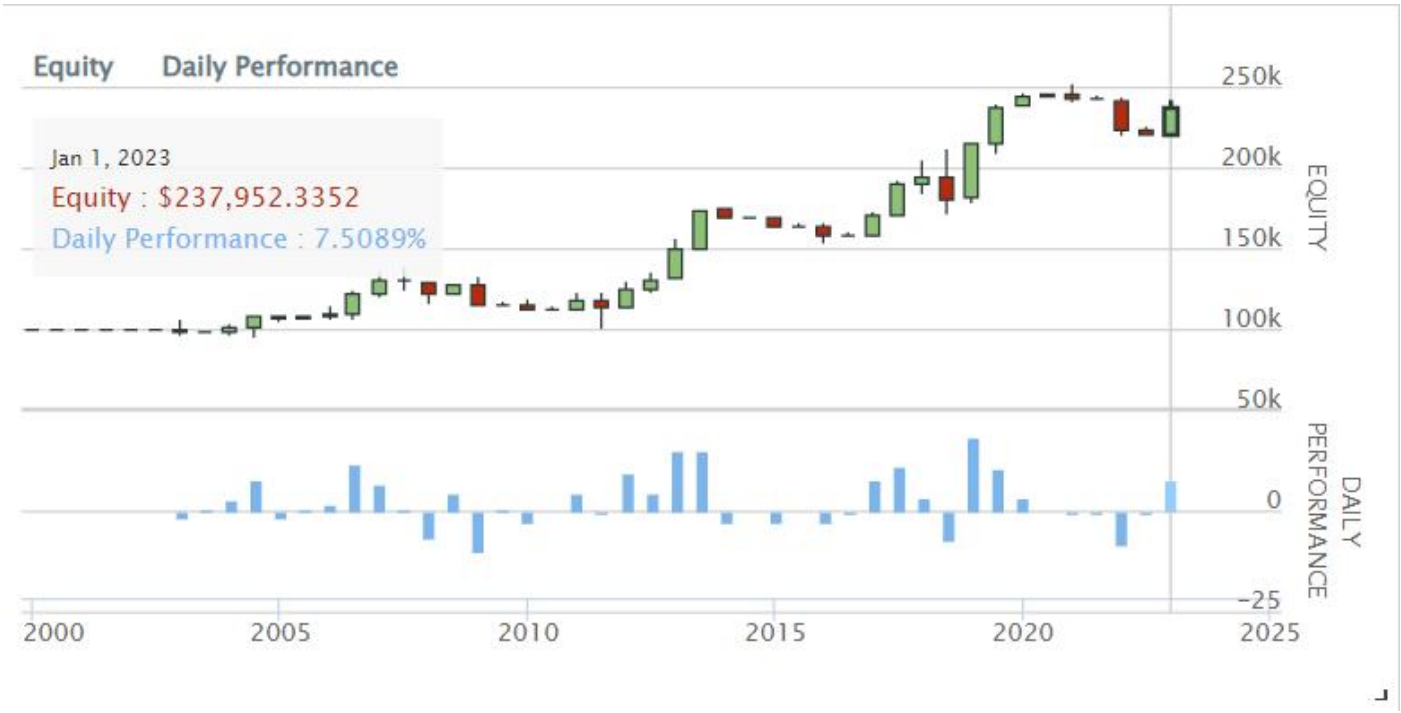

Fig 1. Overall Performance

PSR	0.011%	Sharpe Ratio	0.361
Total Trades	50	Average Win	7.93%
Average Loss	-3.36%	Compounding Annual Return	3.794%
Drawdown	27.700%	Expectancy	1.117
Net Profit	137.952%	Loss Rate	37%
Win Rate	63%	Profit-Loss Ratio	2.36
Alpha	0.016	Beta	0.241
Annual Standard Deviation	0.082	Annual Variance	0.007
Information Ratio	-0.199	Tracking Error	0.142
Treynor Ratio	0.122	Total Fees	$360.77
Estimated Strategy Capacity	$1400000000.00	Lowest Capacity Asset	SPY R735QTJ8XC9X
Portfolio Turnover	0.53%		

Fig 2. Performance Metrics

Fig 3. Drawdown