Not Over Thinking

Option-Expiration Week Effect

Algorithmic Trading Strategy with Full Code

Haixiang

2023.10 | Vol 42.

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

STRATEGY & ECONOMIC RATIONALE

Investors choose stocks from the S&P 100 index as his/her investment universe (stocks could be easily tracked via ETF or index fund). He/she then goes long S&P 100 stocks during the option-expiration week and stays in cash during other days.

BUY	SELL
Goes long S&P 100 stocks during the option-expiration week and stays in cash during other days	The opposite

PARAMETER & VARIABLES

PARAMETER	VALUE
MARKETS TRADED	Equities
FINANCIAL INSTRUMENTS	CFDs, ETFs, futures, stocks
REGION	United States
PERIOD OF REBALANCING	Weekly
NO. OF TRADED INSTRUMENTS	1
WEIGHTING	Equal weighting
LOOKBACK PERIODS	Weekly
LONG/SHORT	Long only

ALGORITHM

```python
from AlgorithmImports import *

class OptionExpirationWeekEffect(QCAlgorithm):

    def Initialize(self):
        self.SetStartDate(2010, 1, 1)
        self.SetCash(10000)

        self.symbol = self.AddEquity("OEF", Resolution.Minute).Symbol

        option = self.AddOption("OEF")
        option.SetFilter(-3, 3, timedelta(0), timedelta(days = 60))

        self.SetBenchmark("OEF")
        self.near_expiry = datetime.min

        self.Schedule.On(self.DateRules.Every(DayOfWeek.Monday, DayOfWeek.Monday),
self.TimeRules.AfterMarketOpen(self.symbol, 1), self.Rebalance)

    def OnData(self, slice):
        if self.Time.date() == self.near_expiry.date():
            self.Liquidate()
    def Rebalance(self):
        calendar = self.TradingCalendar.GetDaysByType(TradingDayType.OptionExpiration, 
self.Time, self.EndDate)
        expiries = [i.Date for i in calendar]
        if len(expiries) == 0: return

        self.near_expiry = expiries[0]

        if (self.near_expiry - self.Time).days <= 5:
            self.SetHoldings(self.symbol, 1)
```

BACKTESTING PERFORMANCE

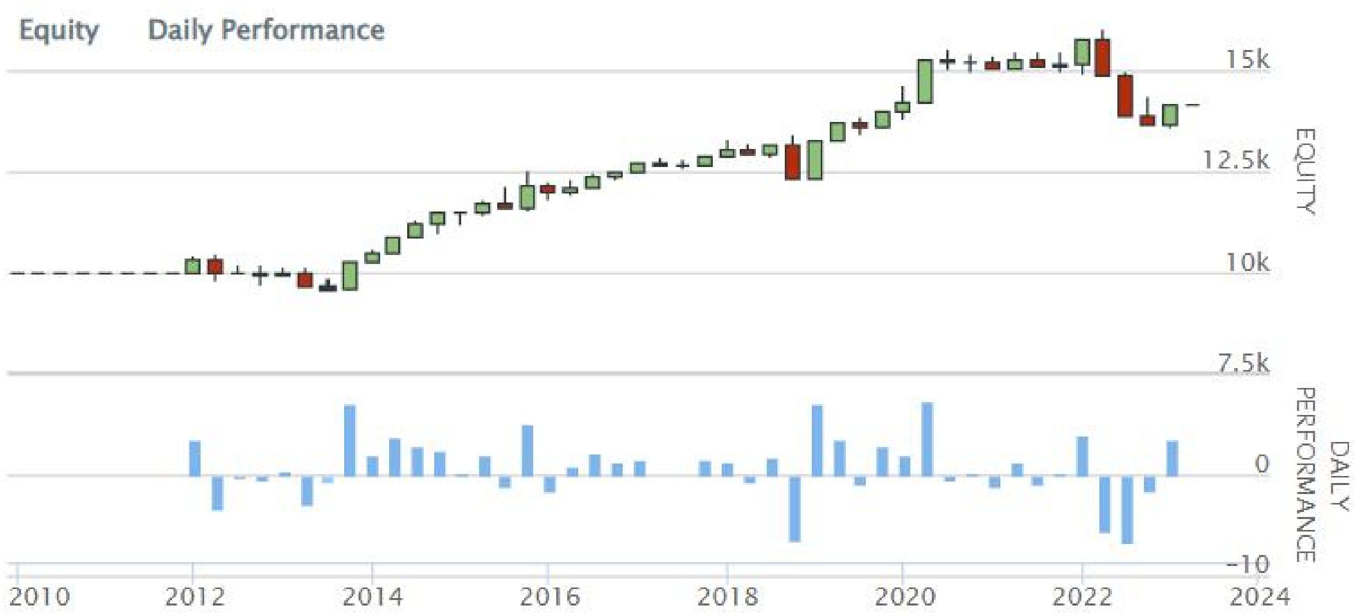
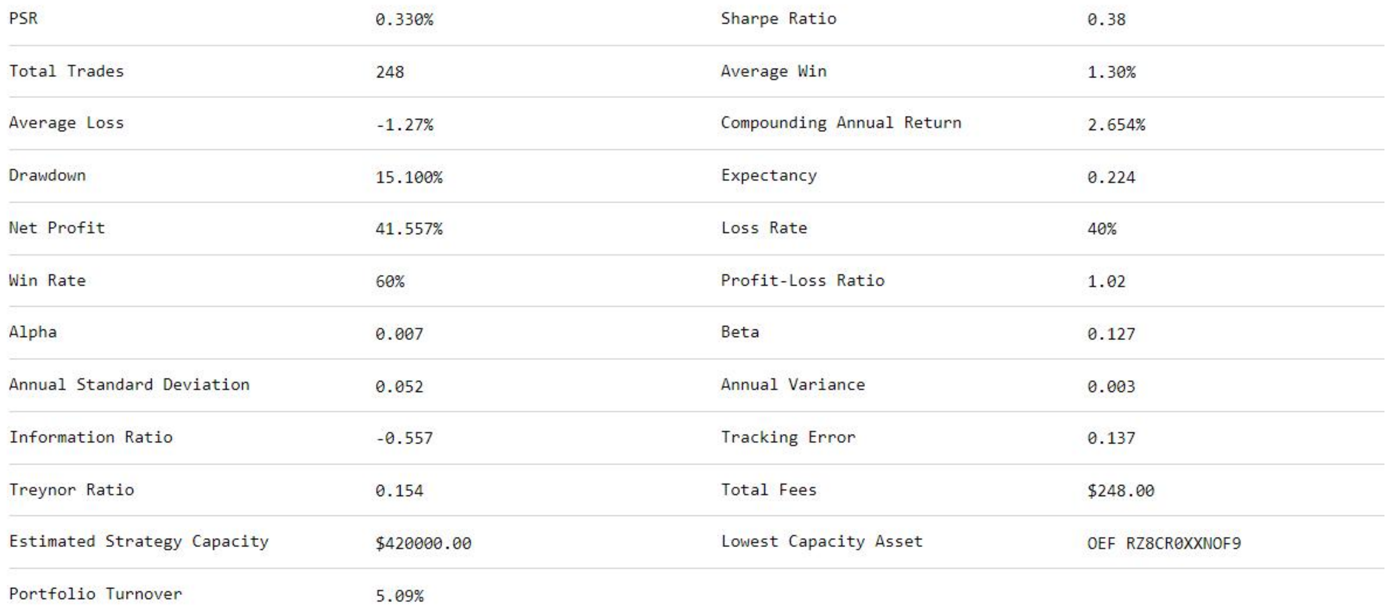


Fig 1. Overall Performance



PSR	0.330%	Sharpe Ratio	0.38
Total Trades	248	Average Win	1.30%
Average Loss	-1.27%	Compounding Annual Return	2.654%
Drawdown	15.100%	Expectancy	0.224
Net Profit	41.557%	Loss Rate	40%
Win Rate	60%	Profit-Loss Ratio	1.02
Alpha	0.007	Beta	0.127
Annual Standard Deviation	0.052	Annual Variance	0.003
Information Ratio	-0.557	Tracking Error	0.137
Treynor Ratio	0.154	Total Fees	$248.00
Estimated Strategy Capacity	$420000.00	Lowest Capacity Asset	OEF RZ8CR0XXNOF9
Portfolio Turnover	5.09%		

Fig 2. Performance Metrics

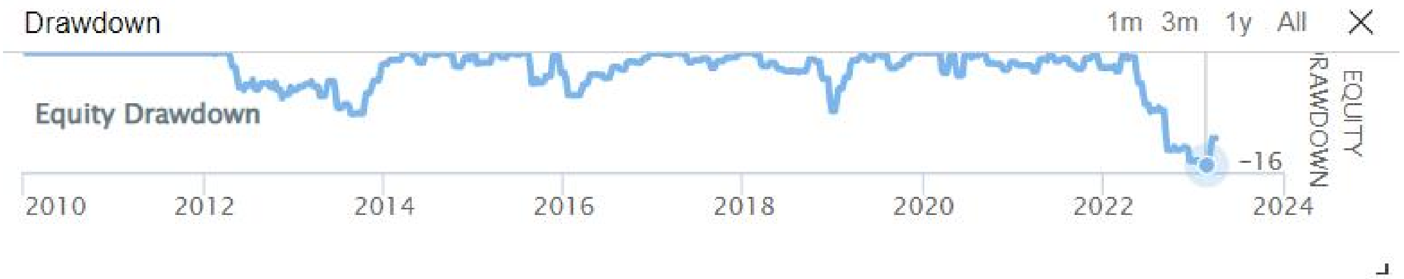


Fig 3. Drawdown