

Option Strategy Evaluation Framework

ETL: (Phase Ia)

- Load historical data from Quandl into Google Cloud
 - <https://www.quandl.com/data/OSMV-ORATS-Smoothed-Options-Market-Quotes/documentation/using-the-data>
 - I will subscribe to this, and furnish you with the logins when appropriate
- Create method to continue loading updated Quandl data daily
 - (This can be run from command line, and manual for V1)
 - V2 Goal would be an automated data load
- Scalability
 - The framework here should be such that it provides for scalability in the granularity of the data; currently it is only EOD, we'd eventually like to move towards hourly, minute, tick.

Interface (Phase Ib)

- There should be an "interface" for testing the strategies with various parameters.
 - This can be "command line" accessible for the initial implementation. I'd simply use it to test various hypotheses manually.
- The "V2" goal of the interface is to be accessible from other applications, whether its a gui on a virtual machine, a web based entry point, mobile, etc.
- Another "V2" goal for the interface would also include an "optimize" button
 - The optimization would look at all of the possible interval/duration/%OTM possibilities, and produce the parameters that yield the greatest PnL.
 - Optimization Constraints
 - Covered calls must be sold above market, i.e. %OTM>0, and Strike>Stock Price
 - Protective Puts must be bought below market, i.e. %OTM>0, and Strike < Stock Price
 - %OTM should not exceed 100%
 - The strategy must make at least 2 trades, so interval cannot be longer than half the period in question
 - There may be more...

Sample Parameters:

Stock	AAPL
Initial Position	1000
Frequency	Monthly
Duration	Monthly
Buy /Sell Option	Sell
Call / Put	Call
% of Position	50%
% OTM	2%
Buy / Sell Stock	(+) 100
Start Period	6/1/2015
End Period	12/31/2016
Commission Rate	\$0.25
Fill Price	Bid

Strategy Logic:

- Covered Calls (First to implement)
 - Covered calls are an option strategy defined by selling calls “against” a long stock position.
 - <https://www.investopedia.com/articles/optioninvestor/08/covered-call.asp>
 - The idea is to sell a number of call options corresponding to the size of the current position, above the current stock price for a certain premium. The profit/loss from the option position at expiration combines with the stock position at expiration.
 - If I am long 1000 shares of AAPL to start, and it is trading at \$165, I would sell up to 10 covered calls at a strike above \$165
 - At the expiration of the option, the stock component will have a PnL and the option component will have a pnl.
 - If I sold 10 of the two week out 170 Call at 1.50, and stock closes at 169 the pnl would be: $(1000*(169-165))+((1.50-0)*100*10)$

- The option would close at 0 on the expiration date, because it expires worthless. If stock was >170, that call would have value, and could even represent a loss.
 - N.B. that this loss above 170 will be exactly offset by gains in the underlying stock.
 - The option change in value is multiplied by 100 because one options contract represents 100 shares.
- The “frequency” parameter will determine how often to implement this strategy.
 - We can make some assumptions about “when” to trade these options. So for example, if options expire on a Friday and the interval/frequency are both one month, we can assume that the option expires at the closing price, and we also simultaneously sell the “next months” call at that closing price.
 - If the start date of the defined period is not the “right day” to start selling one month out options, we should wait until the appropriate date.
 - For example on an option with monthly expirations, if the period started February 1st, the first trade wouldn't be until Feb 16th, to sell the one month out option expiring March 16th, and on March 16th, we'd close that to sell the April 20th expiration.
- The “duration” parameter is the number of days to expiration to target for the option. There will have to be some rounding here, as every class has a different listings calendar (which is apparent in the structure of the data)
 - N.B. In order to maintain the “covered call” ratio, the number of options * 100 can not exceed the shares of stock held. So duration must equal or be less than frequency.
- “% OTM”, this is the parameter used to determine how far out of the money we'd look. So a value of 1%, with AAPL trading \$165, would look for the strike closest to \$166.65
 - Maybe it makes sense to expose a “round up/round down” parameter also?
- Buy/Sell and Call/Put- This would be trading method for the option. To create a covered call strategy it would be “Sell/Call”, and the below protective put strategy would be “Buy/Put”, but my thought was allowing this flexibility outright.
- “Buy/Sell Stock” - This parameter would buy or sell additional shares of stock in the position at the time of option trading. So if you started with 1000 shares, and every month added 100 shares, in addition to your options trading p&l, you would end a 1 year period with 2200 shares $(1000 + 12 * 100)$

- % of Position - This parameter would define the contract count to be used. 100% would be the exact ratio of one option for 100 shares of stock. A parameter of 50% would only sell 5 on 1000 shares. I treat this as a percent, to take into account potential for buying stock above.
 - Start/End Period - The starting/ending dates over which to perform the analysis
 - Commission Rate - this would simply be an additional cost added on for every trade that takes place. I'd also like this to be exposed in the results
 - Fill Price - I'd like to use this parameter to define the price to use in historical PnL analysis. The data includes both a bid and offer price. I'd like the ability to use bid (most likely when selling, i.e. covered calls), the offer (when buying, i.e. protective puts), and also the midpoint (simple average of the two).
- Results
 - The results should break out the PnL from the options, from the stock, from commissions, and have a total
 - For V2, it will also desirable to have standard "stats" on the pnl, like largest drawdown, sharpe ratio, etc.
- Protective Put (Second to implement)
 - A protective put involves buying put contracts as a hedge against a downward move in the stock. If I am long 1000 shares of AAPL at \$165, I would buy
 - The calculations for the pnl are exactly the same, however it would be using long puts, instead of short calls
 - Both options and stock pnl would again be included.
- Additional Strategies
 - While the above two strategies represent the core options strategies I'll be using, I'd also like to design for future flexibility. Treating these strategies modularly, and allowing for additional

Signals / Trading (Phase II)

- The backtest logic should be such that it can generate trading signals. Eventually the plan would be to automate execution for different clients based on their individual trade parameter preferences.
 - I define “trade policy” as the execution strategy for a specific client in a specific name. Clients may have multiple policies in different or the same names.
 - For example, I might have one client that wants to sell AAPL calls 1% above the stock price every week, and another who wants to sell FB calls 5% above the market every other week. Each of those represents a “policy”, that would generate its own independent trade signal at the predetermined interval.
 - While these trading signals are not dependant on ‘what the markets doing’, they take their price from the current market.
 - There should also be flexibility to begin using “no trade” parameters
 - For example, a V2 feature would be adding conditionality to trade signals “sell 1% above market calls in AAPL every week, EXCEPT if volatility is >25%”
- The deliverable for this phase, would be an interface that managed several different trade policies