FINANCE AND ANALYTICS

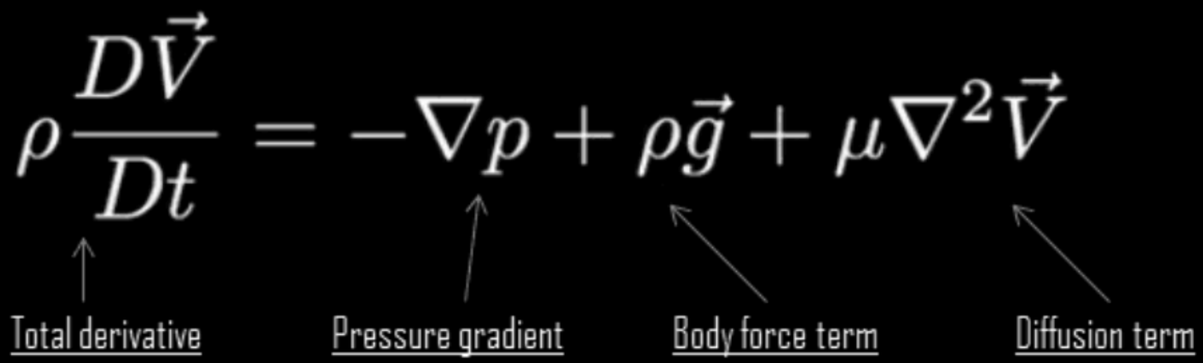
MAKE ME A MARKET

Overview

We divide this problem statement into three parts:

- **Equity Market Making**
  Navier Stokes Stock Price Modeling
- **Options + Equity Arbitrage Market Making**
  Put Call Parity Options Arbitrage
- **Futures + Equity Arbitrage Market Making**
  Cash and Carry Arbitrage

1. Navier Stokes Price Modeling



- We've adapted the Navier-Stokes equation, traditionally used in fluid dynamics, to stock price.
- Our hypothesis is that market dynamics share similarities with fluid behavior, allowing us to leverage this powerful equation for statistical analysis.

1. Navier Stokes Price Modeling

- **Stock Momentum as Fluid Velocity (DV/Dt):**

Price momentum captures the rate and direction of price changes, similar to how velocity describes the motion of fluid particles.

- **Williams %R as Pressure Gradient (-∇p):**

Williams %R measures overbought/oversold conditions, acting like pressure differentials that drive market movements.

- **Chaikin Money Flow (CMF) as Body Force Term (ρg):**

CMF indicates the buying/selling pressure, acting as a 'gravitational' force pulling prices in a certain direction.

- **Density Calculation:**

Density is derived from volume. This gives us a measure of 'market thickness', influencing how easily prices can move.

Behind the Strategy

Fair Price Calculation

Fair Price is calculated using data from previous timestamps by estimating how current market conditions (momentum, sentiment, resistance) evolve over time.

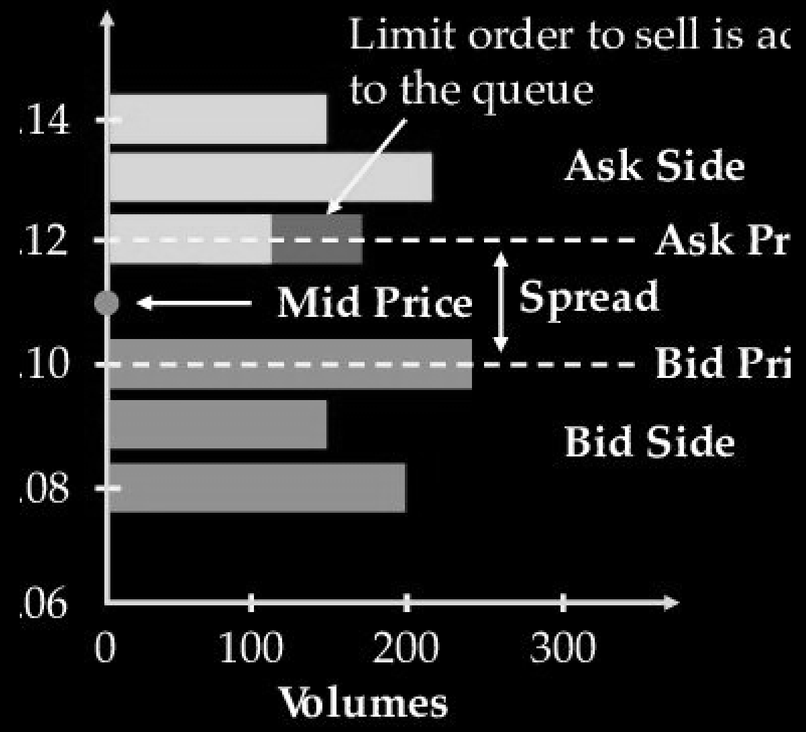
Stock Equilibrium

The model searches for an equilibrium point where forces like bullish and bearish pressures are balanced, calculating a "fair price" where the stock is most likely to stabilize temporarily before the next fluctuation.

Strategy

Bid Price = Volume Weighted Fair Price - Spread

Ask Price = Volume Weighted Fair Price + Spread



These prices create an upper and lower bound around the fair price to execute trades.

Trade Execution

- **Long Trade**

  If the stock's low price in the next candle crosses below the calculated bid price, a long position is initiated.

- **Short Trade**

  If the stock's high price in the next candle crosses above the sell price, a short position is initiated.

The trade is held for a maximum of 15 minutes, or until the exit conditions are met (prices cross opposite bounds).

Risk Management

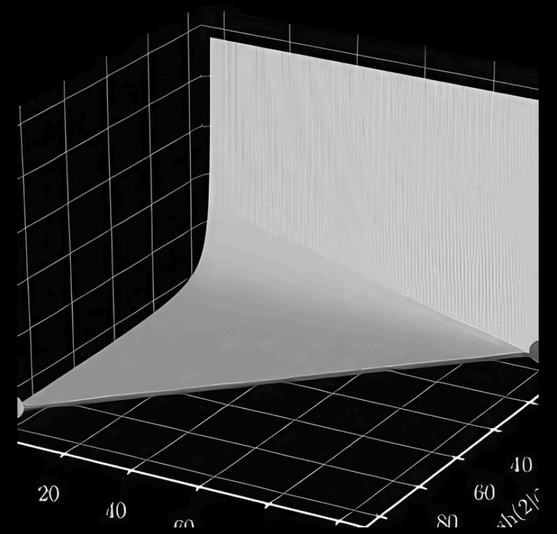
The strategy employs the **Kelly Criterion** to adjust position sizing based on the probability of winning trades and the win-loss ratio. It dynamically calculates the optimal fraction of capital to allocate for each trade, maximizing compounded returns.

Kelly Criterion:

$$f^* = \frac{bp - q}{b}$$

- **f =** fraction of capital to trade
- **b =** win-loss ratio
- **p =** probability of winning
- **q =** the probability of losing

Each trade's returns are compounded based on position size and the Kelly fraction. The strategy records profit/loss for every trade and adjusts capital accordingly.



Varying the Spread

We fix the spread as a % of the stock spot price.

SPREAD	BAJFINANCE	INFOSYS	NIFTY
0.10%	22.93	53.72	53.88
0.25%	475.68	208.37	41.52
0.50%	38.79	30.35	4.36

```
Total PnL: 474.78 %
Winning Trades: 1327
Losing Trades: 569
Sharpe Ratio: 6.159
Sortino Ratio: 11.908
Max Drawdown: 1.67 %
```

*Backtesting done on 1 minute data b/w 2023-01-01 and 2024-01-01

2. Put Call Parity Options Arbitrage

Put-call parity is a fundamental principle in options pricing that establishes a relationship between the prices of **European put options, call options, and their underlying asset**.

$$C + PV(x) = P + S$$

where:

$C =$ Price of the European call option

$PV(x) =$ Present value of the strike price (x), discounted from the value on the expiration date at the risk-free rate

$P =$ Price of the European put

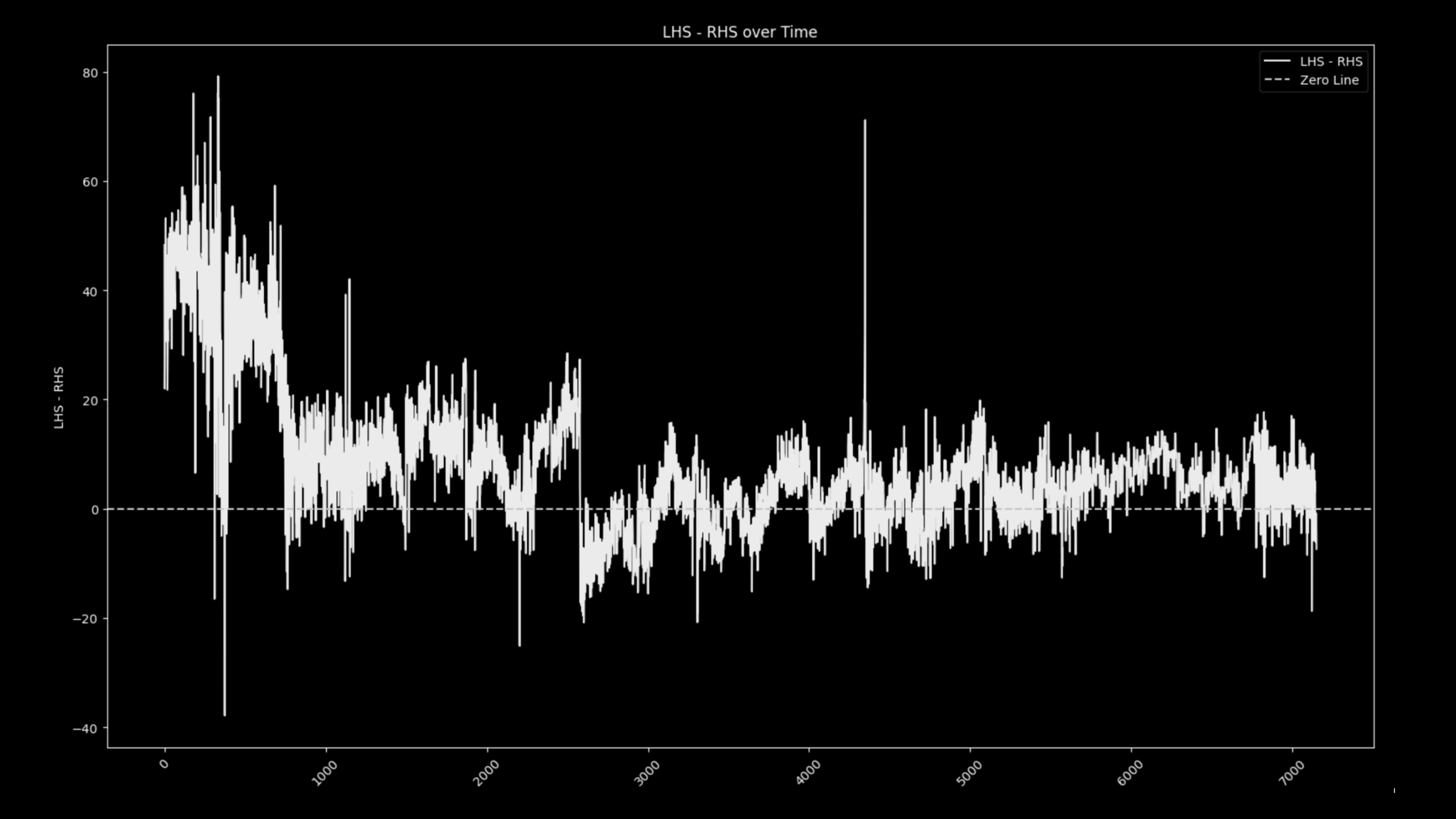
$S =$ Spot price or the current market value of the underlying asset

Violations of this parity relationship may present arbitrage opportunities.

Parity Violation Detection

$$C + PV(x) = P + S$$

- The strategy calculates the **left-hand side (LHS)** and **right-hand side (RHS)** of the put-call parity equation.
- Strategy **entry** is when the absolute difference between LHS and RHS exceeds **10% of the sum of call and put prices**.
- Strategy **exits** is when the absolute difference is less than or equal to **5% of the sum of call and put prices**.


LHS - RHS over Time
LHS - RHS
Zero Line
LHS - RHS
80
60
40
20
0
−20
−40
0
1000
2000
3000
4000
5000
6000
7000

Strategy Execution

- If **LHS > RHS**:
  Short the call, buy the put, and sell the stock.

- If **LHS < RHS**:
  Buy the call, short the call, buy the stock.

Risk Management

Risk Management is done with the help of **Delta Hedging**.
Delta hedging is a critical component of this put-call parity arbitrage strategy. It's used to create a **market-neutral position** when a parity violation is detected, **reducing directional risk** and focusing on profiting from the convergence of option prices to the parity relationship.

Delta Hedging

$$c = S_0\,N(d_1) - K\,e^{-rT}N(d_2)$$

$$p = K\,e^{-rT}\,N(-d_2) - S_0\,N(-d_1)$$

where $$d_1 = \frac{\ln(S_0/K) + (r + \sigma^2/2)T}{\sigma\sqrt{T}}$$

$$d_2 = \frac{\ln(S_0/K) + (r - \sigma^2/2)T}{\sigma\sqrt{T}} = d_1 - \sigma\sqrt{T}$$

- **Calculating Implied Volatility:**

Implied volatility is calculated for both call and put options.
This is done using a root-finding method (**Brent's method**) to solve for the volatility that makes the **Black-Scholes** price equal to the market price.

- **Calculating Delta:**

Using the implied volatilities, the strategy calculates the delta for both options.

- **Position Sizing for Delta Neutrality:**

The strategy then calculates position sizes to achieve delta neutrality.

Position Sizing for Delta Neutrality

- **Stock Quantity = Capital / (3 * Stock Price)**
- **Call Quantity = (Stock Quantity) * abs(Put Delta) / abs(Call Delta)**
- **Put Quantity = Stock Quantity**

  - This approach ensures that the combined delta of the position (stock + call option - put option) is close to zero.

3. Futures-Cash Arbitrage Market Making

Spread = Future Price - Stock Spot Price

- This strategy exploits price discrepancies between the futures market and the underlying cash market. It aims to profit from temporary mispricing while maintaining a delta-neutral position.

Strategy

1. **Enter Position:** If the spread is greater than the 15-minute moving average of spread, we short futures and long stock position.

2. **Exit Position:** If the spread decreases by 50%, the position is exited.

3. **Risk Management**: The delta of a futures contract is 1 (or **-1 when short**), while the **delta of the stock position is 1**. By having equal and opposite positions, the **overall delta becomes zero**.

Data Fetching

All data has been fetched from ICICI Breeze API, which is a publicly available free to use API.

```python
expiry = ["2023-12-28" ,"2023-11-30" ,"2023-10-26" ,"2023-09-28" ,"2023-08-31" ,"2023-07-27" ,"2023-06-28" ,"2023-05-25"

for exp in expiry:

    input_date = datetime.strptime(exp, "%Y-%m-%d")
    first_day_of_current_month = input_date.replace(day=1)
    if input_date.month == 12:
        first_day_of_next_month = input_date.replace(year=input_date.year + 1, month=1, day=1)
    else:
        first_day_of_next_month = input_date.replace(month=input_date.month + 1, day=1)
    first_day_of_current_month_str = first_day_of_current_month.strftime("%Y-%m-%d")
    first_day_of_next_month_str = first_day_of_next_month.strftime("%Y-%m-%d")

    print(exp)

    test = breeze.get_historical_data(interval="1minute",
                                      from_date= f"{first_day_of_current_month_str}T07:00:00.000Z",
                                      to_date= f"{first_day_of_next_month_str}T07:00:00.000Z",
                                      stock_code=ticker,
                                      exchange_code="NFO",
                                      product_type="futures",
                                      expiry_date=f"{exp}T07:00:00.000Z",
                                      right="others",
                                      strike_price="0")

    print(test)

    folder_name = f"{ticker}2023"
    file_name = f"{folder_name}/Futs_{exp}.json"

    os.makedirs(folder_name, exist_ok=True)

    with open(file_name, 'w') as json_file:
        json.dump(test, json_file, indent=4)
```

```python
exp_ranges = {
    "2023-12-28": range(20300, 22100, 100),
    "2023-11-30": range(20300, 22100, 100),
    "2023-10-26": range(20300, 22100, 100),
    "2023-09-28": range(20300, 22100, 100),
    "2023-08-31": range(20300, 22100, 100),
    "2023-07-27": range(20300, 22100, 100),
    "2023-06-28": range(20300, 22100, 100),
    "2023-05-25": range(20300, 22100, 100),
    "2023-04-27": range(20300, 22100, 100),
    "2023-03-29": range(20300, 22100, 100),
    "2023-02-23": range(20300, 22100, 100),
    "2023-01-25": range(20300, 22100, 100)
}
```

```python
for exp, i_range in exp_ranges.items():
    for i in i_range:
        for opt_type in ["call", "put"]:
            strike = i

            input_date = datetime.strptime(exp, "%Y-%m-%d")
            first_day_of_current_month = input_date.replace(day=1)
            if input_date.month == 12:
                first_day_of_next_month = input_date.replace(year=input_date.year + 1, month=1, day=1)
            else:
                first_day_of_next_month = input_date.replace(month=input_date.month + 1, day=1)
            first_day_of_current_month_str = first_day_of_current_month.strftime("%Y-%m-%d")
            first_day_of_next_month_str = first_day_of_next_month.strftime("%Y-%m-%d")


            test = breeze.get_historical_data(interval="1minute",
                                              from_date= f"{first_day_of_current_month_str}T07:00:00.000Z",
                                              to_date= f"{first_day_of_next_month_str}T07:00:00.000Z",
                                              stock_code=ticker,
                                              exchange_code="NFO",
                                              product_type="options",
                                              expiry_date=f"{exp}T07:00:00.000Z",
                                              right=opt_type,
```

Other Approaches

- **Stoikov Model:**

Can be used to set varying spreads instead of fixed percentage of the spot.

Reservation price

$$r(s,q,t) = s - q\gamma\sigma^2(T-t)$$

- **Garman-Klass Estimator:**

Incorporate the open and close prices along with the high and low to refine the estimate of volatility and, indirectly, the spread.

Optimal bid & ask spread

$$\delta^a + \delta^b = \gamma\sigma^2(T-t) + \frac{2}{\gamma}ln(1+\frac{\gamma}{\kappa})$$

- **Corwin-Schultz Estimator:**

Estimates the bid-ask spread using high and low prices over two consecutive trading days


THANK
YOU