

Volume Profile Spread Selection & 2% Return Strategy Guide

A Practical Implementation Guide for Options Traders

Using 120-Day Volume Profile for Strike Selection

Complete Python Implementation
Risk-Defined Strategy Framework
Real-World Examples & Calculations

Guide Date: February 13, 2026

1. Volume Profile Fundamentals for Spread Selection

Volume Profile is a powerful technical analysis tool that displays trading volume distributed across price levels rather than time. Unlike traditional volume indicators that show volume at specific time intervals, Volume Profile reveals where the most trading activity has occurred at specific price levels over a defined period. This information is invaluable for options traders seeking to identify optimal strike prices for credit spreads, as it reveals the market's "fair value" zones and areas of significant support and resistance based on actual trading activity rather than arbitrary technical levels.

1.1 Key Volume Profile Components

Point of Control (POC): The POC represents the price level where the highest volume has been traded during the analysis period. This level indicates where buyers and sellers have found the most agreement on price, making it a significant magnet for future price action. When price moves away from the POC, there is often a tendency for it to be drawn back toward this level, similar to a mean reversion effect. For spread traders, the POC serves as a critical reference point—positions placed beyond the POC have a higher probability of remaining out-of-the-money, while positions placed near the POC face higher risk of being challenged.

High Volume Nodes (HVN): These are price levels or zones where trading volume is significantly higher than surrounding areas. HVN zones represent areas of price acceptance where the market spent considerable time, indicating strong support or resistance depending on whether price is approaching from above or below. When price approaches an HVN zone, it often pauses or reverses, making these levels excellent candidates for strike selection in credit spreads. A put credit spread placed above a significant HVN support zone has the volume profile "backing" the support level.

Low Volume Nodes (LVN): These are price levels where trading volume is significantly lower than surrounding areas. LVN zones represent price rejection areas where the market quickly moved through, indicating that traders did not find value at those prices. Price often moves quickly through LVN zones when revisiting them, making them less reliable for stop placement but useful for understanding market structure. Areas between HVN zones often contain LVN zones, creating a roadmap of price acceptance and rejection.

Value Area (VA): The Value Area represents the price range where approximately 70% of the total volume was traded during the analysis period. This zone typically encompasses the POC and extends above and below it. The upper and lower boundaries of the Value Area (VAH - Value Area High, and VAL - Value Area Low) act as significant support and resistance levels. Credit spreads placed outside the Value Area have statistical backing for remaining successful, as 70% of trading activity historically occurred within the Value Area.

2. Volume Profile Calculation for 120-Day Data

Calculating Volume Profile requires historical price data and a methodology for distributing volume across price levels. There are two primary approaches: using tick-by-tick data (ideal but often unavailable) or using OHLC (Open, High, Low, Close) candlestick data with approximation methods. The latter approach is more practical for most traders and provides sufficiently accurate results for identifying key levels. Below is a complete implementation using the uniform distribution method on daily candlestick data over 120 trading days.

2.1 The Uniform Distribution Method

The uniform distribution method assumes that trading volume is evenly distributed between the high and low of each candle. While this is an approximation, it provides a reasonable estimate of volume distribution across price levels when tick-by-tick data is unavailable. For each trading day, the method distributes the day's volume proportionally across all price bins that fall within the day's high-low range. Over 120 days, this accumulation reveals the volume profile structure and identifies the POC, HVN, and LVN zones with meaningful accuracy.

2.2 Python Implementation

The following Python code demonstrates how to calculate Volume Profile using 120 days of daily OHLC data. The code uses pandas for data manipulation and numpy for numerical operations. The implementation includes functions for calculating the POC, Value Area, High Volume Nodes, and generating strike recommendations for credit spreads.

```
import pandas as pd
import numpy as np
from typing import Tuple, List, Dict

class VolumeProfileAnalyzer:
    """
    Calculate Volume Profile from OHLC data and identify
    key levels for options spread strike selection.
    """

    def __init__(self, num_bins: int = 100):
        self.num_bins = num_bins
        self.profile = None
        self.poc = None
        self.vah = None
        self.val = None
        self.hvn_levels = []
        self.lvn_levels = []

    def calculate_profile(self, df: pd.DataFrame) -> Dict:
        """
        Calculate Volume Profile from OHLC dataframe.

        Args:
            df: DataFrame with columns ['open', 'high', 'low',
                'close', 'volume'] for 120 trading days

        Returns:
            Dictionary with POC, VAH, VAL, HVN, LVN levels
        """
        # Determine price range across all data
```

```

price_min = df['low'].min()
price_max = df['high'].max()
bin_size = (price_max - price_min) / self.num_bins

# Initialize volume profile array
volume_profile = np.zeros(self.num_bins)

# Distribute volume using uniform method
for _, row in df.iterrows():
    high = row['high']
    low = row['low']
    volume = row['volume']

    # Calculate bin indices for this candle's range
    low_bin = int((low - price_min) / bin_size)
    high_bin = int((high - price_min) / bin_size)

    # Ensure within bounds
    low_bin = max(0, min(low_bin, self.num_bins - 1))
    high_bin = max(0, min(high_bin, self.num_bins - 1))

    # Distribute volume uniformly across bins
    num_bins_in_range = high_bin - low_bin + 1
    if num_bins_in_range > 0:
        volume_per_bin = volume / num_bins_in_range
        volume_profile[low_bin:high_bin + 1] += volume_per_bin

# Store profile
self.profile = volume_profile

# Calculate POC (bin with highest volume)
poc_bin = np.argmax(volume_profile)
self.poc = price_min + (poc_bin + 0.5) * bin_size

# Calculate Value Area (70% of volume)
total_volume = volume_profile.sum()
target_volume = total_volume * 0.70

# Expand from POC until 70% volume captured
volume_accumulated = volume_profile[poc_bin]
vah_bin = poc_bin
val_bin = poc_bin

while volume_accumulated < target_volume:
    # Check volumes above and below current range
    vol_above = volume_profile[vah_bin + 1] if vah_bin + 1 < self.num_bins else 0
    vol_below = volume_profile[val_bin - 1] if val_bin - 1 >= 0 else 0

    if vol_above >= vol_below and vah_bin + 1 < self.num_bins:
        vah_bin += 1
        volume_accumulated += vol_above
    elif val_bin - 1 >= 0:
        val_bin -= 1
        volume_accumulated += vol_below
    else:
        break

self.vah = price_min + (vah_bin + 0.5) * bin_size
self.val = price_min + (val_bin + 0.5) * bin_size

# Identify HVN and LVN levels
avg_volume = volume_profile.mean()

```

```

    std_volume = volume_profile.std()

    self.hvn_levels = []
    self.lvn_levels = []

    for i in range(self.num_bins):
        price_level = price_min + (i + 0.5) * bin_size
        vol = volume_profile[i]

        # HVN: volume > mean + 1 std
        if vol > avg_volume + std_volume:
            self.hvn_levels.append({
                'price': price_level,
                'volume': vol,
                'strength': 'strong' if vol > avg_volume + 2*std_volume else 'moderate'
            })
        # LVN: volume < mean - 0.5 std
        elif vol < avg_volume - 0.5 * std_volume:
            self.lvn_levels.append({
                'price': price_level,
                'volume': vol
            })

    return {
        'poc': self.poc,
        'vah': self.vah,
        'val': self.val,
        'value_area_width': self.vah - self.val,
        'hvn_count': len(self.hvn_levels),
        'lvn_count': len(self.lvn_levels),
        'total_volume': total_volume
    }
}

```

2.3 Strike Selection Using Volume Profile

Once the Volume Profile is calculated, the key levels can be used to select optimal strike prices for credit spreads. The strategy involves identifying the "Volume Wall"—a high volume node or the edge of the Value Area that can act as support or resistance—and placing the short strike just beyond this level. This approach provides statistical backing for the strike selection, as the volume profile indicates where the market has historically found value and where price may encounter resistance to further movement.

```
def get_spread_strikes(self, current_price: float,
                      trend: str = 'bullish',
                      min_otm_pct: float = 2.0,
                      spread_width_pct: float = 2.0) -> Dict:
    """
    Determine optimal strike prices for credit spreads
    based on Volume Profile analysis.

    Args:
        current_price: Current underlying price
        trend: 'bullish' for put spreads, 'bearish' for call spreads
        min_otm_pct: Minimum % OTM for short strike (default 2%)
        spread_width_pct: Width of spread as % of underlying

    Returns:
        Dictionary with short_strike, long_strike, and rationale
    """
    if self.poc is None:
        raise ValueError("Run calculate_profile() first")

    # Find nearest HVN level for strike selection
    if trend == 'bullish':
        # Put Credit Spread: Find support below current price
        support_levels = [h['price'] for h in self.hvn_levels
                           if h['price'] < current_price]

        if support_levels:
            # Use highest HVN below current price as support
            volume_wall = max(support_levels)
        else:
            # Fallback to Value Area Low
            volume_wall = self.val

        # Short strike: Below volume wall, min 2% OTM
        min_strike = current_price * (1 - min_otm_pct/100)
        short_strike = min(volume_wall * 0.98, min_strike)

        # Round to nearest strike (assuming 50-point increments)
        short_strike = round(short_strike / 50) * 50

        # Long strike: Spread width below short
        spread_width = current_price * spread_width_pct / 100
        long_strike = short_strike - spread_width
        long_strike = round(long_strike / 50) * 50

        spread_type = 'Bull Put Spread'

    else: # bearish
        # Call Credit Spread: Find resistance above current price
        resistance_levels = [h['price'] for h in self.hvn_levels]
```

```

        if h['price'] > current_price]

    if resistance_levels:
        volume_wall = min(resistance_levels)
    else:
        volume_wall = self.vah

    min_strike = current_price * (1 + min_otm_pct/100)
    short_strike = max(volume_wall * 1.02, min_strike)
    short_strike = round(short_strike / 50) * 50

    spread_width = current_price * spread_width_pct / 100
    long_strike = short_strike + spread_width
    long_strike = round(long_strike / 50) * 50

    spread_type = 'Bear Call Spread'

    # Calculate distance from volume wall
    distance_to_wall = abs(current_price - volume_wall)
    wall_distance_pct = (distance_to_wall / current_price) * 100

    return {
        'spread_type': spread_type,
        'short_strike': short_strike,
        'long_strike': long_strike,
        'volume_wall': volume_wall,
        'poc': self.poc,
        'value_area': (self.val, self.vah),
        'wall_distance_pct': wall_distance_pct,
        'rationale': f"Short strike placed beyond Volume Wall "
                    f"at {volume_wall:.2f}"
    }
}

```

3. Practical Example: 120-Day Volume Profile Analysis

Let's walk through a practical example using a hypothetical F&O; stock to demonstrate how Volume Profile analysis translates into actionable spread recommendations. This example uses simulated data that represents typical patterns seen in liquid Indian F&O; stocks, illustrating the complete workflow from data analysis to strike selection.

3.1 Sample Data Analysis

Scenario: Stock ABC is trading at Rs. 1,850. Analysis of 120 trading days reveals the following Volume Profile structure: Point of Control (POC) at Rs. 1,780, indicating the most accepted price level. Value Area High (VAH) at Rs. 1,820 and Value Area Low (VAL) at Rs. 1,740, representing the 70% volume zone. High Volume Nodes identified at Rs. 1,760 (strong support zone with significant institutional interest), Rs. 1,800 (moderate HVN near current price), and Rs. 1,710 (strong support below VAL). Current price of Rs. 1,850 is above the Value Area High, suggesting the stock is trading in premium territory relative to its 120-day volume profile.

3.2 Bullish Scenario: Put Credit Spread

For a bullish position, we look to sell a Put Credit Spread with the short strike positioned below a Volume Profile support level. The analysis shows: The nearest strong HVN support is at Rs. 1,760,

which is 4.9% below current price. The Value Area Low at Rs. 1,740 provides additional support confirmation. Given the trend is bullish and price is above the Value Area, we can structure a Put Credit Spread with the short strike placed below the Rs. 1,760 HVN support.

Parameter	Value	Rationale
Current Price	Rs. 1,850	Market price
Volume Wall (HVN)	Rs. 1,760	Strong support level
Short Strike	Rs. 1,750	Below HVN, ~5.4% OTM
Long Strike	Rs. 1,700	50-point spread width
Spread Width	Rs. 50	2.7% of underlying
Max Risk	Rs. 50 - Premium	Defined risk
Probability OTM	~75-80%	Based on VP structure

Table 1: Put Credit Spread Structure Using Volume Profile Support

4. Risk-Defined Strategies for 2% Monthly Returns

Achieving a consistent 2% monthly return (approximately 27% annually when compounded) from options trading requires a disciplined approach combining appropriate strategy selection, position sizing, and risk management. This section outlines multiple risk-defined strategies that can be combined to target 2% monthly returns while maintaining strict capital protection through defined-risk structures. The key is to use strategies with high probability of success while managing the risk-reward ratio carefully.

4.1 Strategy 1: Credit Spreads (Primary Strategy)

Structure: Sell an OTM option and buy a further OTM option for protection. For Put Credit Spreads (bullish), sell a put below current price and buy a put at an even lower strike. For Call Credit Spreads (bearish), sell a call above current price and buy a call at a higher strike. The maximum profit is the credit received, and the maximum loss is the spread width minus the credit.

Return Targeting: To achieve 2% monthly return with credit spreads, aim for 1-1.5% return per trade with 2-3 trades per month. Each trade should target approximately 15-25% return on risk capital. For example, with Rs. 100,000 capital and 10% allocation per trade (Rs. 10,000 risk), target Rs. 1,500-2,500 profit per trade. At 2-3 successful trades per month with 70% win rate, this yields approximately Rs. 2,000-3,000 monthly profit (2-3% return).

Parameter	Recommended Range	Notes
Strike Selection	10-15 Delta (OTM)	High probability OTM
Spread Width	2-3% of underlying	Balance risk/reward
Days to Expiry	30-45 days	Optimal theta decay
Profit Target	50% of credit	Take profit early
Stop Loss	2x credit received	Defined exit point
Capital per Trade	5-10% of total	Position sizing

Table 2: Credit Spread Parameters for 2% Monthly Target

4.2 Strategy 2: Iron Condor (Range-Bound Markets)

Structure: An Iron Condor combines a Bull Put Spread and a Bear Call Spread on the same underlying with the same expiration. This creates a position that profits when the underlying stays within a defined price range. The structure involves selling an OTM put and buying a further OTM put (bull put spread) while simultaneously selling an OTM call and buying a further OTM call (bear call spread). Maximum profit is achieved when price stays between the short strikes at expiration.

Return Targeting: Iron Condors are particularly effective for targeting consistent 2% monthly returns when markets are range-bound or have elevated IV. The strategy collects premium from

both sides, potentially doubling the credit compared to a single credit spread. With IV Percentile above 50%, Iron Condors have higher probability of success due to IV mean reversion. Target 2-3% return on capital per Iron Condor, with 1-2 positions per month.

Example: With Nifty at 22,000 and elevated IV ($IVP > 60\%$), construct an Iron Condor: Sell 21,500 Put / Buy 21,400 Put (Bull Put Spread) and Sell 22,500 Call / Buy 22,600 Call (Bear Call Spread). Collect approximately Rs. 100-150 total credit with Rs. 1,000 risk per lot. Probability of profit is approximately 70-75% at expiration. With 5% capital allocation per trade and 70% win rate, target 2% monthly return through disciplined position management.

4.3 Strategy 3: Iron Butterfly (High IV, Low Movement)

Structure: An Iron Butterfly is similar to an Iron Condor but with the short strikes at the same price (typically ATM). This creates a position that profits when the underlying stays very close to the short strike price. The structure offers higher potential returns but with narrower profit zone and higher risk. Best used when IV is very high and you expect minimal price movement, such as after a major event has passed.

Return Targeting: Iron Butterflies can generate higher returns (3-5% per trade) but require more precise market direction assessment. Use only when IV Percentile exceeds 70% and technical indicators suggest range-bound conditions. Allocate smaller capital (3-5%) per trade due to higher risk. This strategy complements credit spreads by providing higher return potential during exceptionally high IV periods.

4.4 Strategy 4: Diagonal Spreads (Directional with Income)

Structure: A Diagonal Spread involves selling a near-term OTM option and buying a longer-term ITM or ATM option of the same type. This creates a position that benefits from time decay on the short leg while maintaining directional exposure through the long leg. The strategy combines elements of covered call writing with defined risk, making it suitable for moderate directional views.

Return Targeting: Diagonal spreads can supplement credit spread income with directional gains. Use when you have a moderate directional view and want to generate income while maintaining exposure. The key advantage is that the long leg can appreciate if the directional move occurs, while the short leg generates income. Target 2-4% return per trade with capital allocation of 5-8% per position.

5. Comprehensive Approach for 2% Monthly Returns

5.1 Portfolio Construction

Rather than relying on a single strategy, a comprehensive approach combines multiple risk-defined strategies to achieve consistent 2% monthly returns while diversifying risk. The allocation should consider current market conditions (IV levels, trend direction, volatility regime) and adjust strategy weights accordingly. Below is a recommended framework for portfolio construction that targets 2% monthly returns with defined risk across all positions.

Strategy	Allocation	Target Return	Market Condition
Credit Spreads	50-60%	1.5-2% per trade	All conditions
Iron Condors	20-30%	2-3% per trade	High IV, range-bound
Iron Butterfly	10-15%	3-5% per trade	Very high IV
Diagonal Spreads	10-15%	2-4% per trade	Directional view
Cash Reserve	20-30%	N/A	Margin buffer

Table 3: Portfolio Strategy Allocation for 2% Monthly Target

5.2 Risk Management Rules

Implementing strict risk management is essential for achieving consistent returns. The following rules should be followed without exception: (1) Maximum capital per trade: 10% of trading capital, with preference for 5-8% allocation. (2) Maximum sector exposure: 25% of total capital to prevent correlated losses. (3) Maximum total deployed capital: 70-80%, maintaining 20-30% cash reserve for adjustments and opportunities. (4) Stop loss execution: Close positions at 2x credit received loss, no exceptions. (5) Profit taking: Close at 50% profit target to lock in gains. (6) Thursday exit rule: Close all positions by 2:30 PM Thursday to avoid physical settlement risk. (7) Drawdown limit: Pause trading if monthly drawdown exceeds 5%.

5.3 Monthly Return Calculation Example

Example with Rs. 100,000 Capital: Following the portfolio allocation, deploy approximately Rs. 70,000 across various strategies while maintaining Rs. 30,000 cash reserve. With credit spreads (Rs. 35,000 allocation, 3 trades), target Rs. 1,750 per winning trade (5% of allocated capital) at 70% win rate. With Iron Condors (Rs. 21,000 allocation, 1-2 trades), target Rs. 600 per winning trade at 75% win rate. Monthly expected return from winning trades approximately Rs. 4,000. Expected losses at 25-30% losing rate approximately Rs. 1,500. Net monthly return approximately Rs. 2,500, representing 2.5% return on total capital. Annual compounding of 2.5% monthly returns yields approximately 34% annual return.

6. Complete Implementation: Volume Profile + Strategy Selection

The following code provides a complete implementation that combines Volume Profile analysis with strategy selection for targeting 2% monthly returns. The implementation includes automatic detection of market conditions, strike selection using Volume Profile, and position sizing based on risk parameters.

```

class StrategySelector:
    """
    Combines Volume Profile analysis with strategy selection
    for targeting consistent monthly returns.
    """

    def __init__(self, capital: float, target_monthly_return: float = 0.02):
        self.capital = capital
        self.target_return = target_monthly_return
        self.vp_analyzer = VolumeProfileAnalyzer(num_bins=100)

    def analyze_and_recommend(self, ohlc_df: pd.DataFrame,
                               current_price: float,
                               iv_percentile: float,
                               trend: str) -> Dict:
        """
        Analyze market and recommend optimal strategy with strikes.

        Args:
            ohlc_df: 120 days of OHLC data
            current_price: Current underlying price
            iv_percentile: Current IV percentile (0-100)
            trend: 'bullish', 'bearish', or 'neutral'

        Returns:
            Strategy recommendation with strikes and sizing
        """
        # Calculate Volume Profile
        vp_data = self.vp_analyzer.calculate_profile(ohlc_df)

        # Determine optimal strategy based on conditions
        if iv_percentile > 70:
            # Very high IV: Iron Butterfly for max premium
            strategy_type = 'iron_butterfly'
            allocation_pct = 0.05 # 5% capital (higher risk)
        elif iv_percentile > 50:
            # Elevated IV: Iron Condor
            strategy_type = 'iron_condor'
            allocation_pct = 0.08 # 8% capital
        else:
            # Normal IV: Credit Spread
            strategy_type = 'credit_spread'
            allocation_pct = 0.10 # 10% capital

        # Get strike recommendations
        if strategy_type == 'iron_condor':
            # Both put and call spreads
            put_strikes = self.vp_analyzer.get_spread_strikes(
                current_price, 'bullish', min_otm_pct=3.0)
            call_strikes = self.vp_analyzer.get_spread_strikes(
                current_price, 'bearish', min_otm_pct=3.0)
            strikes = {'put_side': put_strikes, 'call_side': call_strikes}
        elif strategy_type == 'iron_butterfly':
            # ATM short strikes
            atm_strike = round(current_price / 50) * 50

```

```

        spread_width = current_price * 0.02
        strikes = {
            'short_strike': atm_strike,
            'put_long': atm_strike - spread_width,
            'call_long': atm_strike + spread_width
        }
    else:
        # Single credit spread
        strikes = self.vp_analyzer.get_spread_strikes(
            current_price, trend, min_otm_pct=2.0)

    # Calculate position sizing
    risk_amount = self.capital * allocation_pct
    target_profit = risk_amount * 0.20 # 20% of risk

    return {
        'strategy_type': strategy_type,
        'volume_profile': vp_data,
        'strikes': strikes,
        'position_sizing': {
            'allocation_pct': allocation_pct,
            'risk_amount': risk_amount,
            'target_profit': target_profit,
            'stop_loss_amount': risk_amount * 0.30 # 30% of risk
        },
        'market_conditions': {
            'iv_percentile': iv_percentile,
            'trend': trend,
            'current_price': current_price
        }
    }
}

```

7. Summary and Key Takeaways

Volume Profile analysis provides a data-driven approach to strike selection for options spreads, identifying support and resistance levels based on actual market activity rather than arbitrary technical indicators. The Point of Control, High Volume Nodes, and Value Area boundaries offer statistically-backed levels for positioning credit spreads. By placing short strikes beyond these volume walls, traders can increase the probability of positions remaining profitable.

Achieving consistent 2% monthly returns requires a combination of appropriate strategy selection, disciplined position sizing, and strict risk management. Credit spreads form the foundation of the approach, offering high probability outcomes with defined risk. Iron Condors and Iron Butterflies provide additional opportunities during elevated IV periods. The key is maintaining discipline in following exit rules and not over-allocating capital to any single position.

The Python implementation provided offers a complete framework for analyzing Volume Profile, selecting optimal strikes, and determining appropriate position sizes for targeting 2% monthly returns. Traders should backtest this approach thoroughly using historical data before deploying with real capital, and should always maintain adequate cash reserves for adjustments and unexpected market movements.