Volatility Arbitrage: Backtesting a GARCH-Based Options Trading Strategy

Part 1: Theoretical Foundations of Volatility Trading

This section establishes the conceptual framework for the trading strategy. It details the nature of financial volatility as a tradable asset class, explores its empirical properties, and introduces the quantitative models and trading structures necessary to build a strategy that capitalizes on volatility-based signals.

1.1 The Duality of Volatility: Past vs. Future

At the heart of volatility trading lies the distinction between what has happened and what the market expects to happen. These two perspectives are quantified by historical and implied volatility, respectively.

Historical Volatility (HV) / Realized Volatility

Historical Volatility, also known as Realized Volatility, is a statistical measure of the dispersion of returns for a given security over a specified past period. It is calculated as the annualized standard deviation of logarithmic returns.[1] As a backward-looking metric, HV quantifies the actual price fluctuations an asset has experienced. It serves as a baseline for understanding an asset's inherent riskiness based on its past behavior.

Implied Volatility (IV)

Implied Volatility is a forward-looking measure that captures the market's collective expectation of a security's future volatility.[2] It is not directly observed but is instead derived, or "implied," from the current market price of an options contract. Within the context of an options pricing model like the Black-Scholes-Merton (BSM) model, IV is the value for volatility that, when input into the model, yields a theoretical price equal to the option's current market price.[3] Consequently, IV reflects the consensus view on the potential magnitude of future price swings, heavily influenced by the supply and demand dynamics for options contracts.

The Volatility Spread as a Trading Signal

The core premise of this strategy is that the statistical, backward-looking nature of historical volatility can be used to generate a forecast, which can then be compared against the market's forward-looking implied volatility. The GARCH model, as will be discussed, is a sophisticated method for forecasting future realized volatility based on past return patterns.[4] When the GARCH forecast for future volatility diverges significantly from the market's implied volatility, a potential trading opportunity arises. This "volatility spread" is the fundamental signal for the strategy.

The strategy operates on the hypothesis that a well-specified GARCH model can provide a more accurate forecast of the *actual* volatility that will be realized over the option's life than the current implied volatility. If the market's IV is substantially higher than the GARCH forecast, it suggests that options are "overpriced" from a volatility perspective. A trader can sell these options with the expectation that either the implied volatility will revert downward toward the GARCH forecast, or the actual realized volatility will be lower than what was priced in, allowing the position to profit from the inflated premium. This frames the strategy as a quantitative bet on a superior forecasting model against the market's consensus.

1.2 The Empirical Behavior of Financial Volatility

Financial asset returns exhibit distinct statistical properties that make their volatility somewhat predictable, in stark contrast to the asset prices themselves, which are widely considered to follow a path close to a random walk.

Volatility Clustering

First noted by Benoit Mandelbrot in 1963, volatility clustering is the empirical observation that "large changes tend to be followed by large changes, of either sign, and small changes tend to be followed by small changes".[4] This means that periods of high volatility (large price swings) and periods of low volatility (calm markets) tend to occur in clusters. Statistically, while asset returns themselves show little to no serial correlation, the absolute or squared returns exhibit a positive and slowly decaying autocorrelation function.[4] This clustering phenomenon is a fundamental departure from the assumptions of constant volatility inherent in simpler models and is a key reason why volatility can be forecasted.

Mean Reversion

Volatility is also observed to be mean-reverting. This means that after periods of being unusually high or low, it tends to move back toward its long-term average level.[4] While shocks can cause volatility to spike, these elevated levels are not typically sustained indefinitely; likewise, periods of extreme calm are often followed by a return to more normal levels of price movement. This property provides an anchor for long-term volatility forecasts.

The dual properties of clustering and mean reversion are what make volatility a forecastable time series. Clustering implies that recent volatility is a powerful predictor of near-future volatility. Mean reversion implies that over longer horizons, volatility will be pulled back towards a historical average. A robust volatility forecasting model must therefore account for both of these effects: it needs to be sensitive to recent market shocks while also being anchored to a long-run equilibrium level. This is precisely the architecture of the GARCH model.

1.3 Capturing Volatility Dynamics: The GARCH(1,1) Model

The Generalized Autoregressive Conditional Heteroskedasticity (GARCH) model was developed to formally capture the stylized facts of financial time series, particularly volatility clustering and mean reversion.[4] It is a model of

conditional heteroskedasticity, meaning it models the variance of a time series at a specific point in time, conditional on past information.

Intuition and Specification

The GARCH model posits that today's variance is a weighted average of three components: a long-run average variance, the squared return (or shock) from the previous period, and the variance from the previous period.[5] The most widely used variant is the GARCH(1,1) model, where the "(1,1)" indicates that the model uses one lag of the squared residual and one lag of the past conditional variance.[5]

The model's structure directly addresses the empirical behaviors of volatility:

1. **The ARCH Term (α):** This component links today's variance to the size of yesterday's shock (the squared residual). A large market move yesterday leads to a higher variance forecast for today, capturing the immediate reaction to news or shocks.
2. **The GARCH Term (β):** This component links today's variance to yesterday's variance. A high value for this parameter indicates that volatility is persistent; high volatility yesterday is likely to lead to high volatility today, which mathematically models the phenomenon of volatility clustering.[4]
3. **The Constant Term (ω):** This term is related to the long-run average variance. It acts as the anchor for the mean-reversion process. For a stable GARCH process, the model's long-term variance will revert to $1-\alpha-\beta\omega$.[8]

Because its structure is purpose-built to reflect the observed dynamics of financial markets, the GARCH(1,1) model has proven to be a parsimonious yet remarkably effective tool for volatility forecasting.[7]

1.4 The Volatility Surface: Smile & Skew

A key assumption of the original Black-Scholes-Merton model is that the volatility of the underlying asset is constant across all strike prices and expiration dates. If this were true, the implied volatility derived from options prices would be the same for all options on a given underlying with the same maturity. However, empirical evidence overwhelmingly refutes this. When implied volatility is plotted against strike price for a given expiration, the resulting shape is not a flat line but rather a "smile" or a "skew".[3]

Interpretation of the Smile and Skew

- **Volatility Smile:** This pattern, often seen in currency and near-term equity options, shows higher implied volatility for deep in-the-money (ITM) and far out-of-the-money (OTM) options compared to at-the-money (ATM) options.[3] The U-shape suggests that the market is pricing in a higher probability of large price movements (i.e., "fat tails") in either direction than would be suggested by a lognormal distribution.[11]
- **Volatility Skew (or Smirk):** This is an asymmetric smile, which is the dominant pattern in equity and equity index markets. Typically, implied volatility increases as the strike price decreases, meaning OTM put options have higher IVs than ATM options, which in turn have higher IVs than OTM call options.[10] This "negative skew" reflects the market's greater fear of a sudden market crash (downside risk) than a sudden rally. Consequently, there is higher demand for put options as portfolio insurance, which inflates their prices and, by extension, their implied volatility.[10]

The existence of the volatility smile and skew is a direct reflection of the market's pricing of tail risk and is a well-documented failure of the BSM model's core assumptions.[3] For the purpose of this project, the strategy will focus on ATM options, which represent the lowest point on the smile/skew curve. This is a common simplification for foundational volatility strategies. However, a potential avenue for future improvement is to incorporate the information contained within the entire skew structure.

1.5 Isolating the Volatility Factor: Delta-Neutral Hedging

A strategy that aims to profit from a view on volatility must first neutralize its exposure to the direction of the underlying asset's price. This process is known as delta hedging.

The Concept of Delta (Δ)

Delta is the first of the "Greeks" and measures the rate of change of an option's price with respect to a $1 change in the underlying asset's price.[15] A call option has a delta between 0 and +1, while a put option has a delta between -1 and 0. A long position in 100 shares of the underlying stock has a delta of +100.

The Delta-Neutral Portfolio

A portfolio is considered delta-neutral when the sum of the deltas of all its components is zero.[17] This is achieved by constructing a portfolio of assets with offsetting positive and negative deltas. For example, selling a call option (negative delta) can be hedged by buying a certain number of shares of the underlying stock (positive delta). The goal of a delta-neutral portfolio is to be insensitive to small, incremental movements in the underlying asset's price.[18]

By maintaining a delta-neutral position, a trader effectively strips out the directional component of the portfolio's risk. This isolation is crucial for a pure volatility trading strategy, as it ensures that the portfolio's profit and loss (P&L) is primarily driven by factors other than the direction of the stock price. The main drivers of P&L in a delta-neutral options portfolio become changes in implied volatility (Vega), the passage of time (Theta), and the impact of large price moves (Gamma).[18]

1.6 The Short Straddle: A Canonical Volatility-Selling Strategy

The short straddle is a classic options strategy designed to profit from low volatility and the passage of time. Its structure and risk profile make it a suitable vehicle for executing a view that current implied volatility is overstated.

Structure

A short straddle is constructed by simultaneously selling one at-the-money (ATM) call option and one ATM put option on the same underlying asset, with the same strike price and the same expiration date.[19] The position is established for a net credit, which is the sum of the premiums received from selling both options.

Profit and Loss Profile

The strategy is profitable if, at expiration, the underlying stock price has not moved significantly from the strike price. The maximum profit is limited to the initial net premium received, which occurs if the stock price closes exactly at the strike price at expiration, causing both options to expire worthless.[19] The strategy incurs a loss if the stock price moves beyond one of the two break-even points, which are calculated as the strike price plus or minus the total premium received.[19] The potential loss is theoretically unlimited on the upside and substantial on the downside (down to a stock price of zero).[20]

Exposure to Greeks

The short straddle's P&L is driven by its exposure to the option Greeks:

- **Negative Vega:** A short straddle has negative vega, meaning the position's value increases as implied volatility decreases.[19] This is the central mechanism for profiting from the strategy's core thesis: selling options when implied volatility is high (relative to the GARCH forecast) and anticipating a decline.

- **Positive Theta:** The position has positive theta, meaning it profits from the passage of time.[19] As the expiration date approaches, the time value of both the short call and short put decays, reducing their value and creating a profit for the seller. This provides a steady, positive P&L contribution, assuming the underlying price remains stable.
- **Negative Gamma:** This represents the primary risk of the strategy. A short straddle has negative gamma, which means that its delta is not stable. As the underlying price moves away from the strike, the portfolio's delta changes in an adverse direction.[21] For example, if the stock price rises, the short call's delta becomes more negative faster than the short put's delta moves toward zero, making the overall portfolio delta negative. This requires the trader to buy shares to re-hedge. This dynamic of being forced to "buy high and sell low" to maintain delta neutrality can lead to significant hedging losses during periods of high realized volatility.

Part 2: Mathematical Formulation and Performance Metrics

This section provides the precise mathematical definitions and formulas that form the quantitative backbone of the strategy. It covers the models used for return calculation and volatility forecasting, the option pricing framework, and the standard metrics for evaluating backtest performance.

2.1 Core Model Equations

The following equations are fundamental to the implementation of the volatility arbitrage strategy.

Logarithmic Returns

Logarithmic (or continuously compounded) returns are used for financial time series analysis due to their desirable statistical properties, such as time-additivity. The log return r_t at time t is calculated from the asset price P_t and the previous price P_{t-1}.[23]

$$r_t=\ln\left(\frac{P_t}{P_{t-1}}\right)$$

GARCH(1,1) Model

The GARCH(1,1) model consists of a conditional mean equation and a conditional variance equation. The conditional mean is often modeled as a constant, μ, plus a shock term, u_t.[8] The conditional variance,

σ_t^2, is a function of a constant, the previous period's squared shock, and the previous period's conditional variance.[5]

- Conditional Mean Equation:

  $$r_t=\mu+u_t$$
- Conditional Variance Equation:

  $$\sigma_t^2=\omega+\alpha u_{t-1}^2+\beta\sigma_{t-1}^2$$

  where:
  - σ_t^2: The conditional variance at time t.
  - ω: The constant term (long-run variance intercept).
  - α: The ARCH parameter, which measures the reaction to past shocks.
  - u_{t-1}^2: The squared residual from the mean equation at time $t-1$.
  - β: The GARCH parameter, which measures the persistence of volatility.
  - σ_{t-1}^2: The conditional variance at time $t-1$.

For the model to be stationary, the condition $\alpha+\beta<1$ must hold.[5]

Black-Scholes-Merton (BSM) Model

The BSM model provides a theoretical estimate of the price of European-style options. It is a cornerstone of modern financial theory and will be used to price the options within the backtesting engine.[26]

- European Call Option Price (C):

$C(S0,t)=S0N(d1)-Ke-r(T-t)N(d2)$

- European Put Option Price (P):

  $P(S0,t)=Ke-r(T-t)N(-d2)-S0N(-d1)$

  where:

  $d1=\sigma T-t\ln(S0/K)+(r+2\sigma 2)(T-t)$
  $d2=d1-\sigma T-t$

The "Greeks"

The "Greeks" are measures of the sensitivity of an option's price to changes in various parameters. The formulas below are derived from the BSM model and are essential for risk management.[15]

- **Delta (Δ):** The rate of change of the option price with respect to the price of the underlying asset.
  - Call Delta: $\Delta C=N(d1)$
  - Put Delta: $\Delta P=N(d1)-1$
- Gamma (Γ): The rate of change of Delta with respect to the price of the underlying asset. It is the same for calls and puts.

  $\Gamma=S0\sigma T-tN'(d1)$
- Vega (V): The rate of change of the option price with respect to the volatility of the underlying asset. It is the same for calls and puts.

  $V=S0N'(d1)T-t$
- **Theta (Θ):** The rate of change of the option price with respect to the passage of time (time decay).
  - Call Theta: $\Theta C=-2T-tS0N'(d1)\sigma-rKe-r(T-t)N(d2)$
  - Put Theta: $\Theta P=-2T-tS0N'(d1)\sigma+rKe-r(T-t)N(-d2)$

Table 2.1: BSM & Greeks Variable Definitions

Symbol	Definition	Role in Model

S0	Current price of the underlying asset	Primary input for option value and risk
K	Strike price of the option	Determines the option's moneyness
r	Continuously compounded risk-free interest rate	Discounts the future exercise price
T−t	Time to expiration (in years)	The remaining life of the option contract
σ	Annualized volatility of the underlying asset's returns	Key input representing expected price fluctuation
N(·)	Cumulative distribution function (CDF) of the standard normal distribution	Represents probabilities used in the BSM formula
N′(·)	Probability density function (PDF) of the standard normal distribution	Used in Gamma, Vega, and Theta calculations

2.2 Strategy Performance Evaluation

To objectively assess the performance of the backtested strategy, a set of standardized metrics is required. These metrics evaluate not just the absolute return, but the return relative to the risk taken.

Sharpe Ratio

The Sharpe Ratio measures the average return earned in excess of the risk-free rate per unit of total volatility (standard deviation). It is the most common metric for risk-adjusted return.[28]

Sharpe Ratio=σpE

where:

- E: The expected excess return of the portfolio (Rp) over the risk-free rate (Rf).
- σp: The standard deviation of the portfolio's excess returns.

Sortino Ratio

The Sortino Ratio is a modification of the Sharpe Ratio that differentiates between "good" (upside) and "bad" (downside) volatility. It measures the excess return per unit of downside risk, using the standard deviation of only negative returns in the denominator.[30]

Sortino Ratio=σdE

where:

- σd: The standard deviation of negative portfolio returns (downside deviation).

Maximum Drawdown (MDD)

Maximum Drawdown is a measure of the largest single drop from a portfolio's peak value to a subsequent trough value before a new peak is achieved. It is an indicator of downside risk and quantifies the worst-case loss an investor might have faced over a specific period.[32]

MDD=Peak ValueTrough Value−Peak Value
The result is typically expressed as a negative percentage.

Part 3: Python Implementation: A Step-by-Step Guide

This section provides a practical, step-by-step guide to implementing the volatility arbitrage strategy in Python. It translates the theoretical concepts and mathematical formulas from the preceding sections into executable code. The choice of Python is motivated by its extensive and powerful ecosystem of open-source libraries for data analysis, econometrics, and scientific computing, which makes it ideal for research and backtesting. While languages like C++ offer superior performance for high-frequency execution, Python's ease of use and rapid prototyping capabilities are better suited for a project of this nature.[35]

3.1 Step 1: Data Acquisition and Preparation

The first step is to acquire historical price data for the underlying asset and prepare it for analysis by calculating logarithmic returns.

Python

```python
# Import necessary libraries
import yfinance as yf
import pandas as pd
import numpy as np
from arch import arch_model
from scipy.stats import norm
import matplotlib.pyplot as plt

# --- Step 1: Data Acquisition and Preparation ---

# Define the ticker and the time period for the data
ticker_symbol = 'NVDA'
start_date = '2015-01-01'
end_date = '2023-12-31'

# Download daily historical data using yfinance
try:
    data = yf.download(ticker_symbol, start=start_date, end=end_date)
    print(f"Successfully downloaded {len(data)} rows of data for {ticker_symbol}.")
except Exception as e:
```

```
    print(f"Error downloading data: {e}")
    data = pd.DataFrame() # Create empty dataframe to avoid further errors

# Calculate daily logarithmic returns from the 'Adj Close' prices
# Log returns are preferred for financial time series analysis
if not data.empty:
    data['log_return'] = np.log(data['Adj Close'] / data['Adj Close'].shift(1))
    # Remove the first row which will have a NaN value for log_return
    data = data.dropna()
    print("\nSample of calculated log returns:")
    print(data[['Adj Close', 'log_return']].head())
```

This code block uses the yfinance library to download historical daily data for NVIDIA (NVDA), a notoriously volatile stock, making it a suitable candidate for this strategy.[36] It then calculates the daily log returns, which will serve as the input for the GARCH model.[38]

3.2 Step 2: GARCH Volatility Forecasting

With the returns series prepared, the next step is to fit a GARCH(1,1) model and use it to generate out-of-sample volatility forecasts.

```python
# --- Step 2: GARCH Volatility Forecasting ---

# We will use a rolling window approach to forecast volatility.
# This simulates a real-world scenario where we refit the model periodically.
forecast_horizon = 30
# For demonstration, we'll generate one forecast at a specific point in time.
# A full backtest would loop this process.
split_date = '2022-12-31'
train_data = data[:split_date].log_return * 100 # GARCH models often work better with scaled returns

# Instantiate and fit the GARCH(1,1) model
```

```python
# p=1, q=1 specifies a GARCH(1,1) model.
# vol='Garch' specifies the GARCH volatility process.
# dist='Normal' assumes normally distributed errors.
garch_model = arch_model(train_data, vol='Garch', p=1, q=1, dist='Normal')
model_fit = garch_model.fit(disp='off') # 'disp=off' suppresses convergence output

print("\nGARCH(1,1) Model Summary:")
print(model_fit.summary())

# Generate a multi-step volatility forecast for the next 30 days
forecast = model_fit.forecast(horizon=forecast_horizon, reindex=False)

# Extract, process, and annualize the forecasted variance
forecasted_variance = forecast.variance.values
forecasted_volatility_daily = np.sqrt(forecasted_variance)
# Annualize the volatility by multiplying by the square root of trading days in a year (approx. 252)
garch_forecast_annualized = forecasted_volatility_daily / 100 * np.sqrt(252)

print(f"\nAnnualized GARCH Volatility Forecast for the next {forecast_horizon} days:")
print(garch_forecast_annualized)
```

This snippet uses the arch library to fit a GARCH(1,1) model to the historical log returns.[40] After fitting, it calls the

.forecast() method to predict the conditional variance for the next 30 trading days.[42] The resulting variance is then converted to an annualized volatility, which is the standard convention for quoting volatility and is the required input for the BSM model.

3.3 Step 3: Implied Volatility & Signal Generation

In a real-world application, one would source implied volatility data from a professional data vendor. For this academic project, access to such data is often limited. Therefore, we will simulate a realistic implied volatility series that fluctuates around our GARCH forecast.

Python

```
# --- Step 3: Implied Volatility & Signal Generation ---

# In a real-world scenario, you would fetch ATM implied volatility from an options data provider.
# For this project, we simulate it to be the GARCH forecast plus some random market noise.
# This simulates the market's over- and under-reactions relative to the statistical forecast.
np.random.seed(42) # for reproducibility
noise_factor = 1 + np.random.normal(0, 0.15, size=forecast_horizon)
simulated_iv_annualized = garch_forecast_annualized * noise_factor

# Ensure IV doesn't go below a realistic floor (e.g., 5%)
simulated_iv_annualized = np.maximum(simulated_iv_annualized, 0.05)

# Calculate the "Volatility Spread"
# This is the core signal for our strategy.
# A large positive spread means IV is much higher than our GARCH forecast.
volatility_spread = simulated_iv_annualized - garch_forecast_annualized

print("\nSimulated Implied Volatility vs. GARCH Forecast:")
iv_vs_garch = pd.DataFrame({
    'GARCH_Forecast': garch_forecast_annualized,
    'Simulated_IV': simulated_iv_annualized,
    'Volatility_Spread': volatility_spread
})
print(iv_vs_garch.head())
```

This code generates a simulated_iv_annualized series by adding random noise to the GARCH forecast. This mimics the behavior of market-implied volatility, which is correlated with statistical volatility but also contains its own noise and biases. The volatility_spread is then calculated as the difference between the two, forming the basis for our trading decisions.

3.4 Step 4: Backtesting Engine Logic

The backtesting engine is the core of the project, simulating the strategy's execution over the historical period. The following code outlines the logic for a simplified, event-driven backtester that iterates through the data, checks for trading signals, executes trades, and tracks the portfolio's P&L.

```python
# --- Step 4: Backtesting Engine Logic ---

# First, define the Black-Scholes-Merton pricing functions
def bsm_price(S, K, T, r, sigma, option_type='call'):
    """
    Calculates the price of a European option using the BSM model.
    S: Underlying price
    K: Strike price
    T: Time to expiration (in years)
    r: Risk-free rate
    sigma: Annualized volatility
    """
    d1 = (np.log(S / K) + (r + 0.5 * sigma**2) * T) / (sigma * np.sqrt(T))
    d2 = d1 - sigma * np.sqrt(T)

    if option_type == 'call':
        price = S * norm.cdf(d1) - K * np.exp(-r * T) * norm.cdf(d2)
    elif option_type == 'put':
        price = K * np.exp(-r * T) * norm.cdf(-d2) - S * norm.cdf(-d1)
    return price

# --- Backtesting Loop Logic Outline ---
# This is a conceptual implementation. A full backtester would be more complex.

# Backtest parameters
portfolio = {'cash': 100000, 'position': None, 'pnl':}
pnl_series = pd.Series(index=data.index, dtype=float).fillna(0)
entry_threshold = 0.10 # e.g., trade if IV is 10 percentage points > GARCH
holding_period = 10 # days
risk_free_rate = 0.02 # Assume a constant 2% risk-free rate

# Loop through the trading period (subset of data for demonstration)
backtest_data = data['2023-01-01':].copy()
# In a full backtest, GARCH and IV would be recalculated daily/weekly
# For simplicity, we use the single forecast generated earlier and apply it hypothetically
# to the first potential trade date in our backtest period.
```

```python
is_position_open = False
days_in_trade = 0
trade_details = {}

for date, row in backtest_data.iterrows():
    # --- Check for Entry Condition ---
    # In a real backtest, you'd align the forecast date with the current date
    # Here, we just use the first day of our forecast for demonstration
    current_spread = iv_vs_garch.iloc if not is_position_open else 0

    if not is_position_open and current_spread > entry_threshold:
        is_position_open = True
        days_in_trade = 0

        # Simulate selling a short straddle
        S = row['Adj Close']
        K = S # At-the-money
        T = 30 / 252 # Time to expiration: 30 days
        sigma_entry = iv_vs_garch.iloc

        call_premium = bsm_price(S, K, T, risk_free_rate, sigma_entry, 'call')
        put_premium = bsm_price(S, K, T, risk_free_rate, sigma_entry, 'put')
        premium_received = call_premium + put_premium

        trade_details = {
            'entry_date': date,
            'entry_price': S,
            'strike': K,
            'entry_premium': premium_received,
            'entry_iv': sigma_entry,
            'time_to_expiry': T
        }
        print(f"\n--- TRADE ENTRY on {date.date()} ---")
        print(f"Spread > Threshold ({current_spread:.2f} > {entry_threshold}). Selling Straddle.")
        print(f"Stock Price: ${S:.2f}, Strike: ${K:.2f}, Premium: ${premium_received:.2f}")

    # --- Portfolio Tracking and P&L Calculation for Open Positions ---
    if is_position_open:
```

```
        days_in_trade += 1

        # Recalculate straddle value
        S_current = row['Adj Close']
        T_current = trade_details['time_to_expiry'] - (days_in_trade / 252)
        # Simulate IV decaying towards the GARCH forecast
        iv_current = trade_details['entry_iv'] - (current_spread * (days_in_trade /
holding_period))

        if T_current > 0:
            call_value_current = bsm_price(S_current, trade_details['strike'], T_current,
risk_free_rate, iv_current, 'call')
            put_value_current = bsm_price(S_current, trade_details['strike'], T_current,
risk_free_rate, iv_current, 'put')
            straddle_value_current = call_value_current + put_value_current

            # P&L is the premium received minus the current cost to close the position
            daily_pnl = trade_details['entry_premium'] - straddle_value_current
            pnl_series.loc[date] = daily_pnl

        # --- Check for Exit Condition ---
        if days_in_trade >= holding_period:
            print(f"--- TRADE EXIT on {date.date()} ---")
            print(f"Holding period of {holding_period} days reached. Final P&L: ${daily_pnl:.2f}")
            is_position_open = False
            days_in_trade = 0
            trade_details = {}
            # In a full backtest, we would break the inner loop and continue the outer one
            break # For this simple example, we stop after one trade

print("\nBacktest P&L Series (first 15 days of trade):")
print(pnl_series[pnl_series!= 0].head(15))
```

This code block provides the essential logic for backtesting.[43] It includes a BSM pricing function, defines the entry and exit conditions, and simulates the P&L of an open position day by day. A key feature is the simulation of IV decay, where the market's implied volatility is modeled to revert toward the GARCH forecast over the life of the trade, which is a critical assumption for the strategy's profitability.

3.5 Step 5: Performance Analysis and Visualization

The final step is to analyze the P&L series generated by the backtester and visualize the strategy's performance.

Python

```python
# --- Step 5: Performance Analysis and Visualization ---

def calculate_performance_metrics(pnl_series, risk_free_rate=0.02):
    """Calculates key performance metrics from a daily P&L series."""
    # For this example, we use the P&L from our single trade.
    # A real backtest would have a continuous P&L series.

    # Calculate daily returns from P&L on a hypothetical portfolio value
    daily_returns = pnl_series.diff() / portfolio['cash']
    daily_returns = daily_returns.dropna()

    if len(daily_returns) < 2:
        print("Not enough data to calculate performance metrics.")
        return {}

    # Sharpe Ratio
    excess_returns = daily_returns - (risk_free_rate / 252)
    sharpe_ratio = np.mean(excess_returns) / np.std(excess_returns) * np.sqrt(252)

    # Sortino Ratio
    downside_returns = excess_returns[excess_returns < 0]
    downside_std = np.std(downside_returns)
    sortino_ratio = np.mean(excess_returns) / downside_std * np.sqrt(252) if
downside_std > 0 else np.inf

    # Maximum Drawdown
    cumulative_pnl = pnl_series.cumsum()
```

```
    running_max = cumulative_pnl.cummax()
    drawdown = (cumulative_pnl - running_max) / (running_max + portfolio['cash']) # Pct
of total equity
    max_drawdown = drawdown.min()

    metrics = {
        'Annualized Sharpe Ratio': sharpe_ratio,
        'Annualized Sortino Ratio': sortino_ratio,
        'Maximum Drawdown (%)': max_drawdown * 100
    }
    return metrics

# Calculate metrics for our single trade's P&L
performance = calculate_performance_metrics(pnl_series[pnl_series!= 0])
print("\n--- Performance Metrics ---")
for metric, value in performance.items():
    print(f"{metric}: {value:.2f}")

# Plot the cumulative P&L (Equity Curve)
cumulative_pnl = pnl_series.cumsum()
plt.figure(figsize=(12, 7))
plt.plot(cumulative_pnl.index, cumulative_pnl.values)
plt.title('Strategy Cumulative P&L (Equity Curve)')
plt.xlabel('Date')
plt.ylabel('Cumulative Profit & Loss ($)')
plt.grid(True)
plt.show()
```

This final code block defines a function to compute the Sharpe Ratio, Sortino Ratio, and Maximum Drawdown from the P&L series.[46] It then uses

matplotlib to plot the cumulative P&L, providing a visual representation of the strategy's performance over time, often referred to as the equity curve.[49]

Table 3.1: Backtest Performance Summary

Metric	Value
Total Return (%)	(Calculated from final P&L)

Annualized Sharpe Ratio	(Calculated value)
Annualized Sortino Ratio	(Calculated value)
Maximum Drawdown (%)	(Calculated value)
Total Trades	(Count of trades)
Win Rate (%)	(Percentage of profitable trades)

Part 4: Analysis, Interpretation, and Further Research

This final section provides a critical evaluation of the backtest results, discusses the significant real-world risks and frictions that were simplified in the model, and suggests concrete avenues for future research and strategy improvement.

4.1 Interpreting the Backtest Results

The quantitative metrics generated in the backtest are the primary tools for evaluating the strategy's viability. However, their interpretation requires context.

- **Sharpe Ratio:** A positive Sharpe Ratio indicates that the strategy generated excess returns over the risk-free rate. For a market-neutral strategy like this, which aims to be uncorrelated with broad market movements, a Sharpe Ratio greater than 1.0 is often considered good, while a ratio approaching 2.0 would be considered excellent by institutional standards.[29] It signifies that the strategy delivers strong returns relative to its volatility.
- **Sortino Ratio:** This metric provides a more nuanced view of risk. If the Sortino Ratio is significantly higher than the Sharpe Ratio, it implies that most of the strategy's volatility is on the upside (i.e., large positive P&L swings), which is a desirable characteristic.
- **Maximum Drawdown (MDD):** The MDD is a crucial measure of risk from a practical standpoint, as it represents the worst-case loss an investor would have

historically endured. For a strategy designed to be low-risk, an MDD below 10-15% is generally desirable. A large MDD, even with a high Sharpe Ratio, may make a strategy unpalatable for many investors due to the psychological and financial pain of such a decline.[51]

Ultimately, these metrics must be considered in concert. A strategy with a high Sharpe Ratio but also a high MDD might be suitable for a high-risk fund but not for a conservative one. The ideal strategy exhibits a high Sharpe Ratio, a high Sortino Ratio, and a low MDD, indicating consistent, low-risk returns.

4.2 Real-World Frictions and Risks

The backtest, while informative, is a simplified simulation. In live trading, several risks and frictions can significantly impact performance.

- **Gamma Risk:** This is the most significant risk inherent in any short-volatility strategy like the short straddle. The position has negative gamma, meaning its delta exposure becomes more adverse as the underlying price moves.[22] If the stock price rises, the portfolio's delta becomes increasingly negative; if it falls, the delta becomes increasingly positive. To maintain delta-neutrality, the trader is forced into a "buy high, sell low" hedging pattern. During periods of large, sharp price moves—precisely the kind that can occur around earnings announcements—the accumulated losses from this constant re-hedging (known as "gamma scalping" losses) can easily overwhelm the premium collected from selling the options, leading to substantial losses.[22]
- **Model Risk:** The entire strategy is predicated on the superiority of its models. This introduces two primary sources of model risk [53]:
  1. **GARCH Model Risk:** The GARCH(1,1) model may be misspecified. It might fail to capture certain dynamics of volatility (like the leverage effect), or its parameters might be unstable over time, leading to poor forecasts. If the GARCH forecast is systematically wrong, the entire trading signal is invalid.[55]
  2. **BSM Model Risk:** The BSM model, used for pricing and hedging, relies on assumptions (e.g., lognormal returns, constant volatility, no transaction costs) that are known to be violated in the real world.[14] This means the calculated option prices and hedge ratios are approximations, not certainties.
- **Assignment Risk:** The strategy involves selling American-style options, which can be exercised by the buyer at any time before expiration. Early assignment is

most likely on in-the-money options, particularly on short calls just before an ex-dividend date, as the option holder may exercise to capture the dividend payment.[57] An unexpected assignment can instantly transform a delta-neutral options position into an unwanted long or short stock position, disrupting the portfolio's risk profile and potentially leading to a margin call.[57]

4.3 Avenues for Improvement and Advanced Study

The foundational strategy presented here can be enhanced in several ways to improve its robustness and potential profitability.

- **Advanced Volatility Models:** The standard GARCH(1,1) model assumes a symmetric response to shocks. However, in equity markets, negative shocks (market drops) tend to increase volatility more than positive shocks of the same magnitude—this is the "leverage effect." Models like GJR-GARCH (which adds a term for negative shocks) or EGARCH (Exponential GARCH) are designed to capture this asymmetry and may produce more accurate volatility forecasts, leading to better trading signals.[60]
- **Alternative Options Strategies:** Instead of a short straddle, one could implement a **short strangle**, which involves selling an out-of-the-money call and an out-of-the-money put.[61] This strategy collects less premium but offers a wider range of profitability before losses are incurred, making it potentially more resilient to moderate price moves. It also allows the trader to strategically select strike prices based on the volatility skew, potentially selling options that are "most expensive."
- **Dynamic Signal Thresholds:** The use of a fixed entry threshold (e.g., spread > 0.10) is simplistic. A more sophisticated approach would be to make this threshold dynamic. For example, a trade could be initiated only when the volatility spread is in the top 10% of its historical distribution, ensuring that trades are only taken during periods of truly exceptional (and likely to revert) mispricing.
- **Incorporate the Volatility Skew:** A more advanced version of the strategy could model the entire volatility surface. Instead of just comparing ATM IV to a GARCH forecast, it could analyze the steepness and shape of the skew itself. For instance, a strategy might be designed to sell OTM puts when the negative skew is historically steep, betting on a normalization of crash fears.
- **Regime Filtering:** The strategy's performance may vary significantly under different market conditions. A macro regime filter could be added to improve

performance. For instance, the strategy could be activated only when a broad market volatility index, like the VIX, is above a certain level (e.g., 20). This would focus the strategy on periods when the volatility risk premium is generally higher, potentially improving the risk-reward profile of selling options.

Works cited

1. Figure 7. Implied and Historical Volatility in Equity Markets, accessed July 21, 2025, https://www.imf.org/-/media/Websites/IMF/imported-flagship-issues/external/pubs/ft/GFSR/2010/01/sa/_safigure7pdf.ashx
2. Implied Volatility vs. Historical Volatility: What's the Difference?, accessed July 21, 2025, https://www.investopedia.com/articles/investing-strategy/071616/implied-vs-historical-volatility-main-differences.asp
3. Volatility Skew and Smile - CFA, FRM, and Actuarial Exams Study Notes - AnalystPrep, accessed July 21, 2025, https://analystprep.com/study-notes/cfa-level-iii/volatility-skew-and-smile/
4. Volatility clustering - Wikipedia, accessed July 21, 2025, https://en.wikipedia.org/wiki/Volatility_clustering
5. Building A GARCH(1,1) Model in Python, Step by Step | by Roi ..., accessed July 21, 2025, https://medium.com/@polanitzer/building-a-garch-1-1-model-in-python-step-by-step-f8503e868efa
6. Volatility Clustering in Financial Markets: Empirical Facts and Agent–Based Models - Rama CONT, accessed July 21, 2025, http://rama.cont.perso.math.cnrs.fr/pdf/clustering.pdf
7. What Is the GARCH Process? How It's Used in Different Forms - Investopedia, accessed July 21, 2025, https://www.investopedia.com/terms/g/generalalizedautogregressiveconditionalheteroskedasticity.asp
8. GARCH Estimation – Mingze Gao, accessed July 21, 2025, https://mingze-gao.com/posts/garch-estimation/
9. GARCH Models for Volatility Forecasting: A Python-Based Guide | by The AI Quant | Medium, accessed July 21, 2025, https://theaiquant.medium.com/garch-models-for-volatility-forecasting-a-python-based-guide-d48deb5c7d7b
10. Volatility Smile vs. Skew: Key Differences - LuxAlgo, accessed July 21, 2025, https://www.luxalgo.com/blog/volatility-smile-vs-skew-key-differences/
11. Volatility Skew: How it Can Signal Market Sentiment - Investopedia, accessed July 21, 2025, https://www.investopedia.com/terms/v/volatility-skew.asp
12. The Volatility Smile and Skew - YouTube, accessed July 21, 2025, https://www.youtube.com/watch?v=d3sRfwyS1g4
13. Volatility Skew - Definition, Types, How it Works - Corporate Finance Institute, accessed July 21, 2025, https://corporatefinanceinstitute.com/resources/derivatives/volatility-skew/

14. What Is a Volatility Smile? - SoFi, accessed July 21, 2025, https://www.sofi.com/learn/content/volatility-smile/
15. Option Greeks: The 4 Factors to Measure Risk - Investopedia, accessed July 21, 2025, https://www.investopedia.com/trading/getting-to-know-the-greeks/
16. Option Greeks Calculator: Delta, Gamma, Theta, and Vega ..., accessed July 21, 2025, https://tradingblock.com/calculators/option-greeks-calculator
17. iongroup.com, accessed July 21, 2025, https://iongroup.com/blog/markets/delta-neutral-trading-is-a-growing-requirement-for-buy-sides-how-can-brokers-respond/#:~:text=In%20essence%2C%20delta%2Dneutral%20trading,a%20hedge%20against%20market%20volatility.
18. Delta Neutral: Definition, Use With a Portfolio, and Example, accessed July 21, 2025, https://www.investopedia.com/terms/d/deltaneutral.asp
19. Short Straddle - Fidelity, accessed July 21, 2025, https://www.fidelity.com/learning-center/investment-products/options/options-strategy-guide/short-straddle
20. Short Straddle Payoff and Break-Even Points - Macroption, accessed July 21, 2025, https://www.macroption.com/short-straddle-payoff/
21. Gamma Options: What is Gamma in Options Trading? - tastylive, accessed July 21, 2025, https://www.tastylive.com/concepts-strategies/gamma
22. What Is Gamma in Investing and How Is It Used? - Investopedia, accessed July 21, 2025, https://www.investopedia.com/terms/g/gamma.asp
23. What Are Logarithmic Returns and How to Calculate Them in ..., accessed July 21, 2025, https://saturncloud.io/blog/what-are-logarithmic-returns-and-how-to-calculate-them-in-pandas-dataframe/
24. Log Return - Rate of Return Expert, accessed July 21, 2025, https://www.rateofreturnexpert.com/log-return/
25. GARCH(1,1) - frds, accessed July 21, 2025, https://frds.io/algorithms/garch/
26. The Black-Scholes-Merton Model | AnalystPrep - FRM Part 1 Study ..., accessed July 21, 2025, https://analystprep.com/study-notes/frm/part-1/valuation-and-risk-management/the-black-scholes-merton-model/
27. The Black-Scholes Model, accessed July 21, 2025, https://www.columbia.edu/~mh2078/FoundationsFE/BlackScholes.pdf
28. corporatefinanceinstitute.com, accessed July 21, 2025, https://corporatefinanceinstitute.com/resources/career-map/sell-side/risk-management/sharpe-ratio-definition-formula/#:~:text=Sharpe%20Ratio%20%3D%20(Rx%20%E2%80%93%20Rf)%20%2F%20StdDev%20Rx&text=Rx%20%3D%20Expected%20portfolio%20return,portfolio%20return%20(or%2C%20volatility)
29. Sharpe Ratio - How to Calculate Risk Adjusted Return, Formula, accessed July 21, 2025, https://corporatefinanceinstitute.com/resources/career-map/sell-side/risk-management/sharpe-ratio-definition-formula/
30. Sortino Ratio | Formula + Calculator - Wall Street Prep, accessed July 21, 2025, https://www.wallstreetprep.com/knowledge/sortino-ratio/

31. Sortino Ratio - Overview, How To Calculate, When To Use - Corporate Finance Institute, accessed July 21, 2025, https://corporatefinanceinstitute.com/resources/wealth-management/sortino-ratio-2/
32. Max Drawdown Definition and Formula - YCharts, accessed July 21, 2025, https://ycharts.com/glossary/terms/max_drawdown
33. Maximum Drawdown (MDD): Definition and Formula - Investopedia, accessed July 21, 2025, https://www.investopedia.com/terms/m/maximum-drawdown-mdd.asp
34. Peak-to-Valley Drawdown - Investopedia, accessed July 21, 2025, https://www.investopedia.com/terms/p/peak-to-valley-drawdown.asp
35. Python vs C++ for HFT | Best Programming Language for Algo ..., accessed July 21, 2025, https://www.youtube.com/watch?v=DSdbm4CBdqA
36. yfinance Library - A Complete Guide - AlgoTrading101 Blog, accessed July 21, 2025, https://algotrading101.com/learn/yfinance-guide/
37. Download Financial Dataset Using Yahoo Finance in Python | A Complete Guide, accessed July 21, 2025, https://www.analyticsvidhya.com/blog/2021/06/download-financial-dataset-using-yahoo-finance-in-python-a-complete-guide/
38. pandas-ta/pandas_ta/performance/log_return.py at main - GitHub, accessed July 21, 2025, https://github.com/twopirllc/pandas-ta/blob/main/pandas_ta/performance/log_return.py
39. python - Calculating log-returns across multiple securities and time, accessed July 21, 2025, https://quant.stackexchange.com/questions/21092/calculating-log-returns-across-multiple-securities-and-time
40. Introduction to ARCH Models - arch 7.2.0, accessed July 21, 2025, https://arch.readthedocs.io/en/stable/univariate/introduction.html
41. ARCH Modeling - arch 7.2.0, accessed July 21, 2025, https://arch.readthedocs.io/en/latest/univariate/univariate_volatility_modeling.html
42. Forecasting - arch 7.2.0, accessed July 21, 2025, https://arch.readthedocs.io/en/latest/univariate/forecasting.html
43. optopsy - PyPI, accessed July 21, 2025, https://pypi.org/project/optopsy/1.0.1/
44. Backtesting.py – An Introductory Guide to Backtesting with Python - Interactive Brokers, accessed July 21, 2025, https://www.interactivebrokers.com/campus/ibkr-quant-news/backtesting-py-an-introductory-guide-to-backtesting-with-python/
45. Algorithmic-Trading---Backtesting---Banknifty-Straddle-using-Python ..., accessed July 21, 2025, https://github.com/umeshpalai/Algorithmic-Trading---Backtesting---Banknifty-Straddle-using-Python/blob/main/straddle_backtest_bnf.ipynb
46. Sharpe Ratio Explained: Formula, Calculation in Excel & Python ..., accessed July 21, 2025, https://blog.quantinsti.com/sharpe-ratio-applications-algorithmic-trading/

47. Sharpe, Sortino and Calmar Ratios with Python | Codearmo, accessed July 21, 2025, https://www.codearmo.com/blog/sharpe-sortino-and-calmar-ratios-python
48. How To Calculate The Drawdown In Python For Your Trading ..., accessed July 21, 2025, https://www.quantifiedstrategies.com/how-to-calculate-trading-drawdown-in-python/
49. Cumulative percentage of a column in Pandas - Python ..., accessed July 21, 2025, https://www.geeksforgeeks.org/python/cumulative-percentage-of-a-column-in-pandas-python/
50. How to Plot a Cumulative Distribution Function CDF of a Pandas Series in Python, accessed July 21, 2025, https://saturncloud.io/blog/how-to-plot-a-cumulative-distribution-function-cdf-of-a-pandas-series-in-python/
51. Maximum Drawdown - Overview, Investment Risk, Portfolios - Corporate Finance Institute, accessed July 21, 2025, https://corporatefinanceinstitute.com/resources/career-map/sell-side/capital-markets/maximum-drawdown/
52. Gamma - The Options Industry Council, accessed July 21, 2025, https://www.optionseducation.org/advancedconcepts/gamma
53. Model risk | Futures of quantitative finance - Université Paris Cité, accessed July 21, 2025, https://futures-quantitative-finance.u-paris.fr/research-axis/model-risk/
54. Model Risk: Definition, Management, and Examples - Investopedia, accessed July 21, 2025, https://www.investopedia.com/terms/m/modelrisk.asp
55. Model risk - Wikipedia, accessed July 21, 2025, https://en.wikipedia.org/wiki/Model_risk
56. Navigating Model Risk in Finance, accessed July 21, 2025, https://www.numberanalytics.com/blog/navigating-model-risk-in-finance
57. Options Assignment: Understanding the Risks and Opportunities ..., accessed July 21, 2025, https://www.barchart.com/education/options-assignment-education
58. The Risks of Options Assignment | Charles Schwab, accessed July 21, 2025, https://www.schwab.com/learn/story/risks-options-assignment
59. Options Assignment Explained: How Does Assignment Work? - tastylive, accessed July 21, 2025, https://www.tastylive.com/concepts-strategies/assignment
60. GARCH vs. GJR-GARCH Models in Python for Volatility Forecasting - QuantInsti Blog, accessed July 21, 2025, https://blog.quantinsti.com/garch-gjr-garch-volatility-forecasting-python/
61. The Benefits of Systematically Selling Volatility - RiverPark Funds, accessed July 21, 2025, https://www.riverparkfunds.com/assets/pdfs/news/Structural_Alpha_White_Paper_Final.pdf