Empirical Analysis of the Black–Scholes Model and Volatility Estimation Methods

Lucia de Lorenzo-Caceres Cana

July 2025

1 Abstract

This project applies the Black–Scholes model to real market data from Apple Inc. to evaluate how different volatility estimation methods affect option pricing accuracy. Using Python, the study compares arbitrary, historical, bootstrapped, and implied volatility through root mean square error (RMSE) analysis. Results show that implied volatility provides the most accurate prices, while bootstrapping offers only a minor improvement over historical estimates. Overall, the model performs well even with constant volatility assumptions, highlighting its robustness and practical value in financial modeling.

2 Introduction

This paper builds upon a collaborative university project focused on the mathematical derivation of the Black-Scholes formula for option pricing. Expanding on that foundation, the present work shifts from purely theoretical analysis to practical application within real financial markets. It incorporates Python programming to implement and test various pricing models, with particular attention given to the role of volatility in option valuation.

Specifically, the paper investigates how different approaches to estimating volatility, such as using historical data or assuming stochastic volatility, affect the accuracy of the Black-Scholes model. It also explores the numerical computation of implied volatility and examines the behaviour of the "Greeks" to better understand the model's sensitivities. Understanding how volatility estimation impacts the accuracy of a model is important for model calibration and understanding the performance of a model. This paper aims to determine which volatility estimation method yields the most accurate options prices under the Black-Scholes framework.

3 Introduction to Options

An **option** is a financial derivative that gives the holder the right, but not the obligation, to buy or sell an underlying asset at a predetermined **strike price** within a specified time frame. There are two primary types of options:

- **American Options**: Can be exercised at any time before or on the expiration date.
- **European Options**: Can only be exercised on the expiration date.

The **strike price** is the agreed price at which the holder may buy or sell the asset. The **stock price** refers to the current market value of the asset at the time of exercise. The **premium** is the amount paid by the buyer to acquire the option contract. [1]

3.1 Call vs. Put Option Pricing

A **call option** gives the holder the right to *buy* the underlying asset. It is profitable to exercise a call option if the market price of the asset exceeds the strike price, allowing the holder to purchase at a lower price and potentially sell at market value for a gain. If the market price is lower than the strike price, the option remains not exercised.
A **put option** grants the holder the right to *sell* the underlying asset. It is beneficial to exercise a put option if the market price is below the strike price, enabling the holder to sell at a higher-than-market price. If the market price is above the strike price, the option is not exercised. In both types of options, there

is a buyer and a seller. However, the risk and reward profiles vary depending on the option type and the role of the party involved [1].

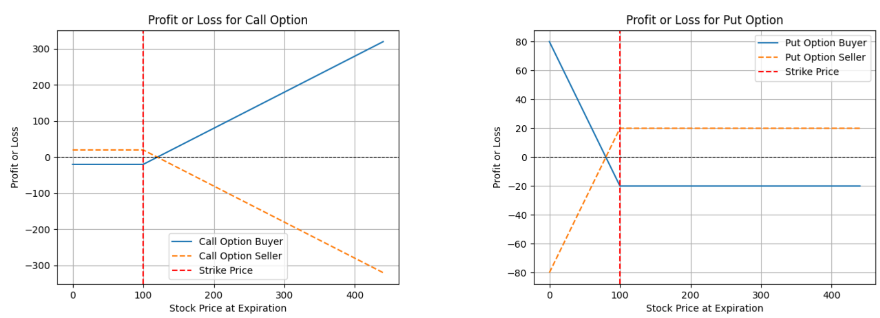
3.2 Call Option Payoff

- **Buyer**: Maximum loss is limited to the premium paid. Potential profit is theoretically *unlimited* as the price of the underlying asset can rise without bound.
- **Seller**: Maximum gain is limited to the premium received. However, losses can be *unlimited* if the asset price increases significantly.

3.3 Put Option Payoff

- **Buyer**: Loss is limited to the premium paid. Maximum profit is achieved if the underlying asset's price falls to zero.
- **Seller**: Profit is limited to the premium received, while potential losses can be substantial if the asset's price drops significantly.

These payoff structures can be more clearly understood by visualizing the profit and loss graphs for both buyers and sellers of call and put options.



(a) Call Option Payoff

(b) Put Option Payoff

Figure 1: Payoff diagrams for call and put options

4 Black-Scholes Model

The Black-Scholes model, developed in 1973 by Fischer Black and Myron Scholes and later extended by Robert Merton, is a foundational framework in financial mathematics. It provides a closed, form solution for pricing European options. The model is based on the principle that a hedged portfolio, constructed by going long in one asset and short in another, should earn the risk-free rate of return, thereby eliminating arbitrage opportunities [3].
However, the formula relies on several assumptions:

- The option is European and can only be exercised at expiration.
- There are no transaction costs for buying or selling the option.
- The short-term risk-free interest rate is known and constant.
- The underlying stock pays no dividends during the option's life.
- The stock price follows a log-normal distribution.
- Time is continuous.
- Volatility is constant and does not vary with time.
- Short selling is allowed with no constraints or penalties.

Under these assumptions, the price of a European call option is given by the **Black-Scholes formula**:

$$\begin{aligned} V(S,t) &= SN(d_1) - Ke^{-r(t^*-t)}N(d_2) \\ d_1 &= \frac{\ln\left(\frac{S}{K}\right) + (r + 0.5\sigma^2)(t^* - t)}{\sigma\sqrt{t^* - t}} \\ d_2 &= d_1 - \sigma\sqrt{t^* - t} \end{aligned}$$

where $w(S,t)$ is the value of the call option at time t, S is the current price of the underlying asset, K is the strike price, r is the risk-free interest rate, σ is the volatility of the underlying asset, t^* is the expiration time, and $N(\cdot)$ is the cumulative distribution function of the standard normal distribution [1].

4.1 Assessing the model

The model accuracy would be would be assessed by using the root mean square error (RMSE) it measures the average magnitude of the error providing a measurement of error in the same unit as the price, the goal is to minimize the RMSE.

The RMSE can be computed using the formula,

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

where n is the total number of observation, y_i is the actual price of the market, $\hat{y}_i$ is the price predicted by the model.

4.2 Implementing the Formula in Python

To implement the Black-Scholes formula in Python, I created a function that accepts all components of the model as input parameters. I used the SciPy library to access the cumulative distribution function of the standard normal distribution, avoiding the need to implement it manually. The function also includes conditional statements to handle invalid inputs, such as zero or negative volatility or maturity, ensuring the code runs even in the presence of erroneous inputs.

To verify the implementation, I tested a hypothetical example using the following parameters: $S = 100$, $t = 0$, $K = 80$, $r = 0.01$, $\sigma = 0.25$, and $t^* = 30$ days. The resulting call option price was approximately \$63.11. Since the strike price is \$80 and the current stock price is \$100, the option is *in the money* and would be exercised at expiration, resulting in a profit of \$20 per share assuming there are no transaction costs.

Next, I applied the implementation to real market data for **Apple Inc. (AAPL)** using the `yfinance` Python package. The spot price was obtained from AAPL's most recent daily closing price. The expiration date was chosen as the first available from the ticker's option chain. Since the Black-Scholes formula requires the time to maturity as a continuous value in years, I converted the number of days until expiration to a fraction of a year using 252 trading days per year.

I then extracted the available strike prices from the call option chain and applied the Black-Scholes formula to each using fixed values for the risk-free interest rate ($r = 0.05$) and volatility ($\sigma = 0.25$). This generated a set of theoretical call option prices, which were plotted alongside actual market prices to assess model performance.

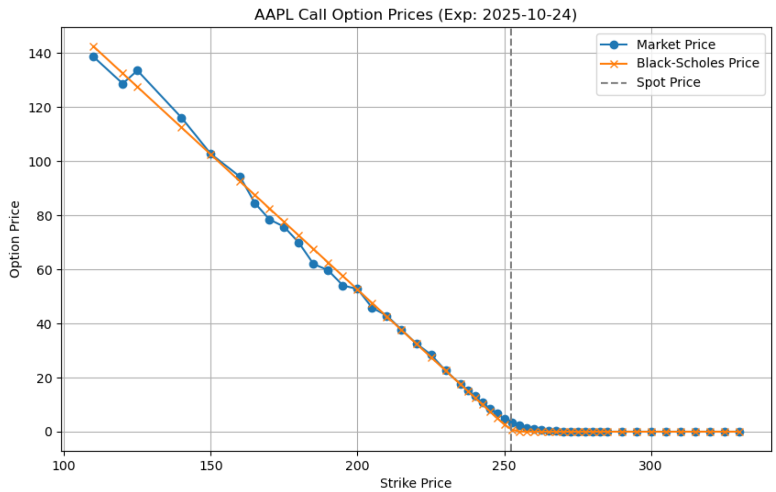


Figure 2: Model vs market prices ($\sigma = 0.05$)

The plots in Figure 2 indicate that the Black–Scholes model provides a reasonable approximation of call option prices. The model's predictions generally follow the trend of observed market prices, although some deviations appear for strikes further from the current stock price. The calculated RMSE of $1.979 corresponds to an average deviation of approximately 6.3% relative to the mean market price. This level of error is within a reasonable range for simple pricing models, suggesting a fairly good overall fit. However, the discrepancies observed are primarily attributable to the assumption of constant volatility, which does not hold in real markets. Moreover, since the volatility parameter was chosen arbitrarily, it may not accurately reflect actual market conditions. The error could be reduced by selecting a volatility that better represents the market, such as one estimated from historical data.

4.3 Historical Volatility

Historical volatility is a statistical measure of how much an asset's price has fluctuated over a specific time period. It is commonly calculated as the standard deviation of daily returns, providing insight into the asset's past price stability. A higher historical volatility suggests greater uncertainty and risk [2].
In Python, historical volatility can be computed by:

- Calculating daily returns using `pct_change()`.
- Applying a rolling window using `rolling(window = n)`.
- Computing the standard deviation within each window using `std()`.

- Annualizing the result by multiplying by $\sqrt{252}$.

Using three months of daily closing prices for Apple and a 30-day rolling window, the historical volatility was estimated to be approximately 25.42%. This volatility differs significantly from the arbitrary volatility used in the previous example. Implementing this value in the Black-Scholes model as before.

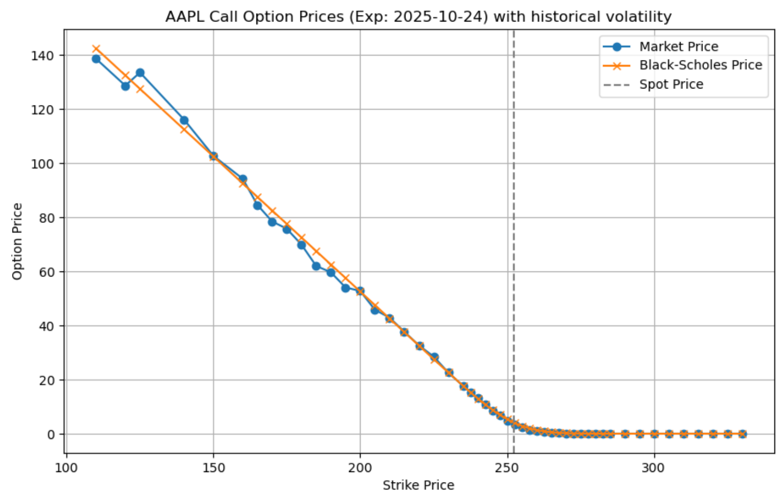


Figure 3: Model vs market prices (historical volatility)

In Figure **??**, we observe that using historical volatility slightly improves accuracy especially near the at-the-money strike prices. However, despite the 20% difference in volatility the graphs visually similar, suggesting that the constant-volatility assumption is not the dominant source of error for prices in this dataset. Quantitatively, the RMSE falls from 1.979 to 1.8629, an absolute reduction of 0.1166 (about a 6% decrease). Although this is a measurable improvement, it is modest, indicating that realized (historical) volatility alone does not fully capture the market's expectation of future volatility. This is because we are trying to estimate the present volatility with past volatility although it could potentially be a good estimator it will have significant error as historical data does not include other factors that affect the volatility of a current market. To decrease the effect of this factors we can use bootstrapping.

4.4 Bootstrapping Historical Volatility

Bootstrapping is a resampling method used to mimic the sampling process without requiring new data. In this case, we resample from the existing market prices, using a sample size equal to the original dataset. For each resample, the historical volatility is computed and stored. This process is repeated 5,000 times to produce a sampling distribution of volatility estimates.

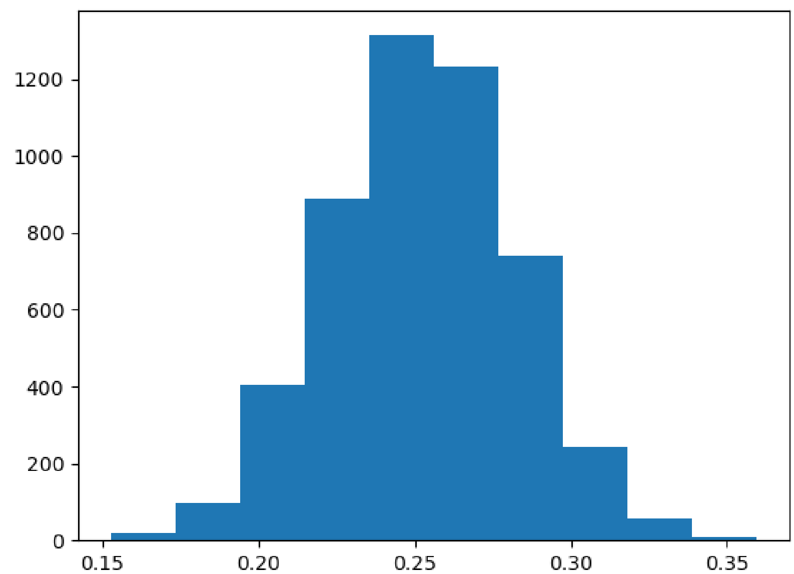


Figure 4: Sampling distribution of historical volatilities

From this sampling distribution, a 95% confidence interval can be constructed by taking the 2.5th and 97.5th percentiles of the bootstrap distribution. Hence, the confidence interval for volatility is $[0.1946, 0.3072]$. This means we can be 95% confident that the true market volatility lies within this range. In other words, if this resampling procedure were repeated many times, approximately 95% of the calculated intervals would contain the true volatility.

To assess whether bootstrapping provides a better estimate than the standard historical volatility, we can consider the centre of the bootstrap distribution. Since the distribution appears approximately normal and symmetric, the mean serves as an appropriate measure of central tendency. The mean of the bootstrap sampling distribution gives a volatility estimate of 25.15%. This represents a small 0.27% difference compared to the directly computed historical volatility. Furthermore, the root mean square error (RMSE) decreases slightly to 1.8615, a reduction of 0.014. Although this improvement suggests that bootstrapping yields a marginally better estimate, the gain may not justify the additional computational effort.

4.5 Implied Volatility

Implied volatility represents the market's forecast of a stock's future volatility and is derived from current option prices. Unlike historical volatility, which is backward-looking, implied volatility is forward-looking and incorporates market sentiment, expectations, and risk premiums.

To compute implied volatility, the Black-Scholes formula is used in reverse: we input the market price of an option and solve for the volatility that would yield that price. However, the formula is not algebraically invertible for volatility, so numerical methods must be employed [4].

One commonly used method is Newton-Raphson iteration, defined as:

$$x_n = x_{n-1} - \frac{f(x_{n-1})}{f'(x_{n-1})}$$

where f is the difference between the Black-Scholes price and the market price, and f' is the derivative of that function with respect to volatility.

Before diving into implied volatility calculations, it is beneficial to understand the option Greeks, sensitivity measures that describe how option prices respond to changes in market conditions.

4.6 Greeks

The option Greeks are sensitivity measures that describe how the price of an option responds to changes in underlying parameters. They are essential tools in option risk management and pricing. The five key Greeks are: Delta, Gamma, Vega, Theta, and Rho.

- **Delta (Δ)**: Measures the rate of change of the option price with respect to changes in the underlying asset's price. It estimates how much the option price is expected to move for a small change in the stock price.

  $$\Delta = \frac{\partial V}{\partial S}$$

  where V is the option price and S is the underlying asset price.

- **Gamma (Γ)**: Measures the rate of change of Delta with respect to the underlying asset's price. It reflects how Delta itself changes as the underlying price changes.

  $$\Gamma = \frac{\partial^2 V}{\partial S^2}$$

- **Vega**: Measures the sensitivity of the option price to changes in volatility. It indicates how much the option price is expected to change with a 1%

change in the asset's volatility.

$$\text{Vega} = \frac{\partial V}{\partial \sigma}$$

where σ is the volatility of the underlying asset.

Deriving Delta

We start with the Black-Scholes formula for the price of a European call option: To compute Delta, we differentiate V with respect to S:

$$\Delta = \frac{\partial V}{\partial S}$$

Applying the product and chain rules:

$$\Delta = \frac{\partial}{\partial S}\left[SN(d_1)\right] - \frac{\partial}{\partial S}\left[Ke^{-r(T-t)}N(d_2)\right]$$

Differentiating each term:

$$\frac{\partial}{\partial S}\left[SN(d_1)\right] = N(d_1) + S \cdot N'(d_1) \cdot \frac{\partial d_1}{\partial S}$$

$$\frac{\partial}{\partial S}\left[Ke^{-r(T-t)}N(d_2)\right] = Ke^{-r(T-t)} \cdot N'(d_2) \cdot \frac{\partial d_2}{\partial S}$$

Since $\frac{\partial d_1}{\partial S} = \frac{\partial d_2}{\partial S} = \frac{1}{S\sigma\sqrt{T-t}}$, we can write:

$$\Delta = N(d_1) + S \cdot N'(d_1) \cdot \frac{1}{S\sigma\sqrt{T-t}} - Ke^{-r(T-t)} \cdot N'(d_2) \cdot \frac{1}{S\sigma\sqrt{T-t}}$$

Simplifying:

$$\Delta = N(d_1) + \frac{N'(d_1)}{\sigma\sqrt{T-t}} - \frac{Ke^{-r(T-t)}N'(d_2)}{S\sigma\sqrt{T-t}}$$

In practice, the additional terms are often negligible for estimation purposes, so we approximate:

$$\Delta \approx N(d_1)$$

Deriving Gamma

Since Gamma is the second derivative of the option price with respect to S, we compute:

$$\Gamma = \frac{\partial^2 V}{\partial S^2} = \frac{\partial \Delta}{\partial S}$$

Only the term involving $N(d_1)$ contributes significantly:

$$\Gamma = \frac{\partial}{\partial S}\left[N(d_1)\right] = N'(d_1) \cdot \frac{\partial d_1}{\partial S}$$

Using the earlier result $\frac{\partial d_1}{\partial S} = \frac{1}{S\sigma\sqrt{T-t}}$, we get:

$$\Gamma = \frac{N'(d_1)}{S\sigma\sqrt{T-t}}$$

Deriving Vega

To compute Vega, we differentiate V with respect to volatility σ:

$$\text{Vega} = \frac{\partial V}{\partial \sigma}$$

Applying the chain rule:

$$\text{Vega} = S \cdot N'(d_1) \cdot \frac{\partial d_1}{\partial \sigma} - Ke^{-r(T-t)} \cdot N'(d_2) \cdot \frac{\partial d_2}{\partial \sigma}$$

We compute:

$$\frac{\partial d_1}{\partial \sigma} = \frac{\sqrt{T-t}}{2} - \frac{\ln(S/K) + (r + \frac{1}{2}\sigma^2)(T-t)}{\sigma^2\sqrt{T-t}}$$

But with algebraic simplification, the full expression for Vega becomes:

$$\text{Vega} = SN'(d_1)\sqrt{T-t}$$

This is the standard closed-form used in most applications.

Now that we understand the Greeks to some extent, we can use **Vega** to calculate the **implied volatility**, as by definition, Vega is the derivative of the call option price with respect to volatility. Hence, in Newton's method, Vega serves as the derivative f'.
Since we know that for the Black-Scholes formula,

$$\text{Vega} = SN'(d_1)\sqrt{T-t},$$

we can create a function in Python to compute Vega. The parameters of the function are S (spot price), t (current time), K (strike price), r (risk-free rate), ω (volatility), and t^* (maturity). Inside the function, d_1 is computed using the same formula as in the Black-Scholes model. Then, Vega is calculated using the normal probability density function $N'(d_1)$, which can be implemented using the `scipy.stats.norm.pdf` function.

Furthermore, we define a function to compute **implied volatility** using **Newton's method**, where the parameters are: b (the market call option price), S, K, t, r, σ_0 (initial guess σ), t^*, tolerance for convergence and maximum number of iterations.

The function iterates up to the specified maximum number of iterations. In each iteration, it:

1. Computes the Black-Scholes price using the current estimate of volatility,
2. Computes the corresponding Vega,

3. Evaluates the function $f(\omega) = \text{BS\_price}(\omega) - b$, and applies Newton's method to update ω.

When this implied volatility function is applied to Apple (AAPL) stock options, it returns a list of implied volatilities, along with `NaN` values where the method does not converge. To select a specific volatility value for modeling, we:

1. Remove all `NaN` values from the list,
2. Use the `np.median()` function to compute the median implied volatility.

This results in an implied volatility estimate for Apple stock of approximately 23.56%. Using this implied volatility in the Black-Scholes model, rather than historical volatility, yields the following graphs:

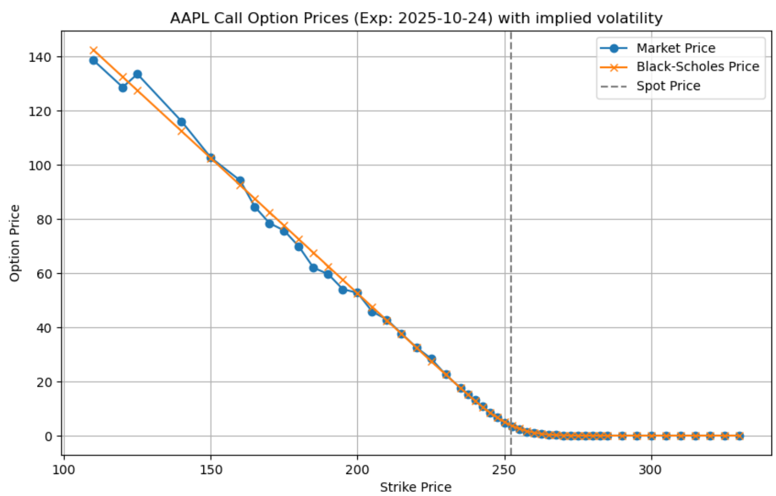


Figure 5: Model vs market prices (implied volatility)

From Figure 4, it is evident that there is an improvement in accuracy, particularly for options priced closer to the spot price. This suggests that the volatility estimated through bootstrapping is closer to the true market volatility of Apple's stock. Furthermore, the root mean square error (RMSE) decreases to 1.8561, reinforcing the improvement observed in the plot.

Nevertheless, some discrepancies remain, which are likely attributable to the simplifying assumptions of the Black–Scholes model, such as a constant interest rate and volatility. It is important to note that any model serves as an approximation of reality, no model can be 100% accurate. The goal is to achieve a reasonable estimate of market behavior, and some level of error is therefore expected.

5 Conclusion

Table 1: Your Table Caption

Method	Volatility (%)	RMSE
Random	5	1.9794
Historical	25.41	1.8629
Bootstrap	25.15	1.8615
Implied	23.56	1.8561

In conclusion, the Black–Scholes model provides a reasonably accurate estimate of market prices. However, the choice of the volatility parameter plays an important role in reducing model error. In this study, we found that using implied volatility yields the lowest error, followed by bootstrapped historical volatility. Interestingly, even when a random volatility outside the confidence interval was used, the model still produced a relatively good approximation of market prices. This finding suggests that deviations from the true market volatility do not drastically affect the model's performance, indicating that the assumption of constant volatility—though simplified—may not have a major impact on the accuracy of the Black–Scholes price estimation. Future work could involve investigating if implied volatility also provides the most accurate results in other pricing models.