Derivation of Black-Scholes Formula with Applications in Pricing and Hedging

Ethan Scott
30117295

February 28, 2024

Abstract

The Black-Scholes formula is a mathematical model used to calculate the theoretical price of European-style options. The formula takes into account factors such as the current stock price, the option's strike price, time to expiration, risk-free interest rate, and volatility to determine the fair value of the option. By providing a method for pricing options, the Black-Scholes formula has become a cornerstone of modern financial theory and has numerous applications in derivative pricing, risk management, and investment strategy.

1 Definitions

1.1 Brownian Motion

Brownian motion is a stochastic process used to model random movements, such as particles in a fluid or gas. In finance, it serves to represent the random fluctuations in asset prices over time. Mathematically, Brownian motion is characterized by its continuous paths and constant variance, making it a useful tool for modeling the uncertainty inherent in financial markets. In the context of option pricing, Brownian motion is often used to model the stochastic behavior of underlying asset prices, influencing the dynamics of option prices. The expected value of Brownian motion is 0, and its variance increases linearly with time.

Formally, a stochastic process $\{X(t), t \geq 0\}$ is said to be a Brownian motion process if it satisfies the following properties:

(i) $X(0) = 0$,

(ii) $\{X(t), t \geq 0\}$ has stationary and independent increments,

(iii) For every $t > 0$, $X(t)$ is normally distributed with mean 0 and variance $\sigma^2 t$.

[Ros23]

1.1.1 Geometric Brownian Motion

Geometric Brownian motion (GBM) is a type of Brownian motion commonly used in financial modeling to describe the stochastic behavior of asset prices. It is characterized by a constant drift term and a volatility term, reflecting the continuous compounding of returns over time. Geometric Brownian motion is a key component of the Black-Scholes model and is instrumental in pricing derivative securities such as options.

A stochastic process $\{Y(t), t \geq 0\}$, is a Geometric Brownian motion process with drift coefficient μ and variance parameter σ^2, then the process $\{X(t), t \geq 0\}$ defined by $X(t) = e^{Y(t)}$.[Ros23]

1.2 Standard Normal Distribution

The standard normal distribution, also known as the Gaussian distribution, is a probability distribution that describes the variation of a random variable around its mean. It has a bell-shaped curve symmetric about the mean, with a mean of 0 and a standard deviation of 1. The cumulative distribution function of the standard normal distribution, denoted as $N(\cdot)$, plays a crucial role in option pricing models such as the Black-Scholes model. It provides probabilities associated with specific values or ranges of values of a normally distributed random variable, facilitating the calculation of option prices and risk measures.

1.3 Risk-Neutral Pricing

Risk-neutral pricing is a fundamental concept in financial mathematics that forms the basis for derivative pricing models such as the Black-Scholes model. It assumes that investors are indifferent to risk and value assets based on their expected future payoff discounted at the risk-free rate. In risk-neutral pricing, the expected return on an asset is adjusted using the risk-free interest rate to account for the time value of money and the uncertainty associated with future cash flows. This approach allows for the consistent valuation of derivative securities and enables the derivation of option pricing formulas that reflect market expectations under a risk-neutral framework.

2 Derivation

2.1 European Call Options

The Black-Scholes Formula for pricing a European Call option is given by:

$$C^E(0) = S(0)N(y_1) - Ke^{-rT}N(y_2) \tag{1}$$

where $C^E(0)$ is the call option price at time $t = 0$, $S(0)$ is the current stock price, K is the strike price, r is the risk-free interest rate, T is the time to maturity, and $N(\cdot)$ is the cumulative distribution function of the standard normal distribution.

To derive this formula, we start with the expectation of the payoff of a European Call option at maturity:

$$C^E(0) = e^{-rT}E_Q[\max\{S(T) - K, 0\}] \tag{2}$$

where $S(T)$ is the stock price at maturity. By the risk-neutral pricing principle, the expected payoff discounted at the risk-free rate is the option price.[CZ11]

We express $S(T)$ using geometric Brownian motion:

$$\begin{aligned} S(T) &= S(0)e^{(r-\frac{\sigma^2}{2})T+\sigma W_T} \\ &= S(0)e^{\sigma W_T - \frac{\sigma^2 T}{2}}, \end{aligned}$$

where σ is the volatility and W_T is a standard Brownian motion at time T.

We will now derive Equation (1) using Equation (2). We know that:

$$E_Q[e^{-rT}S(T)] = S(0), \text{ and}$$

$$E[e^{\sigma W_T - \frac{\sigma^2 T}{2}}] = 1. \text{ [CZ11]}$$

So,

$$C^E(0) = e^{-rT}\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} \max\{S(0)e^{(r-\frac{\sigma^2}{2})T-\sigma\sqrt{T}y} - K, 0\}e^{\frac{-y^2}{2}}dy. \tag{3}$$

From Equation (3), it follows that:

$$S(0)e^{(r-\frac{\sigma^2}{2})T-\sigma\sqrt{T}y} = K \to (r - \frac{\sigma^2}{2})T - \sigma\sqrt{T}y = \ln(\frac{K}{S(0)})$$

$$\to y = \frac{\ln(\frac{K}{S(0)}) - (r - \frac{\sigma^2}{2})T}{\sigma\sqrt{T}} \approx y_0.$$

Then,

$$\begin{aligned}
C^E(0) &= e^{-rT}\frac{1}{\sqrt{2\pi}}\int_{y_0}^{\infty} S(0)(e^{(r-\frac{\sigma^2}{2})T-\sigma\sqrt{T}y} - K)e^{\frac{-y^2}{2}}\,dy \\
&= \frac{1}{\sqrt{2\pi}}\int_{y_0}^{\infty} S(0)e^{-\frac{\sigma^2 T}{2}-\sigma\sqrt{T}y} * e^{-\frac{y^2}{2}} - e^{-rT}Ke^{-\frac{y^2}{2}}\,dy \\
&= \frac{1}{\sqrt{2\pi}}\int_{y_0}^{\infty} S(0)e^{-\frac{y^2}{2}-\frac{\sigma^2 T}{2}-\sigma\sqrt{T}y} - e^{-rT}Ke^{-\frac{y^2}{2}}\,dy.
\end{aligned}$$

From this integral, we can combine the first exponential so that:

$$e^{-\frac{y^2}{2}-\frac{\sigma^2 T}{2}-\sigma\sqrt{T}y} = e^{-\frac{y^2-\sigma^2 T-2\sigma\sqrt{T}y}{2}} = e^{\frac{-(y-\sigma\sqrt{T})^2}{2}}.$$

So,

$$\begin{aligned}
C^E(0) &= \frac{1}{\sqrt{2\pi}}\int_{y_0}^{\infty} S(0)e^{\frac{-(y-\sigma\sqrt{T})^2}{2}}\,dy - e^{-rT}K\frac{1}{\sqrt{2\pi}}\int_{y_0}^{\infty} e^{-\frac{y^2}{2}}\,dy \\
&= \frac{1}{\sqrt{2\pi}}\int_{y-\sigma\sqrt{T}}^{\infty} S(0)e^{-\frac{x^2}{2}}\,dx - e^{-rT}K[N(+\infty) - N(y_0)] \\
&= \frac{S(0)}{\sqrt{2\pi}}\int_{y-\sigma\sqrt{T}}^{\infty} e^{-\frac{x^2}{2}}\,dx - e^{-rT}KN(-y_0) \\
&= S(0)[1 - N(y_0 - \sigma\sqrt{T})] - e^{-rT}KN(-y_0) \\
&= S(0)N(\sigma\sqrt{T} - y_0) - e^{-rT}KN(-y_0),
\end{aligned}$$

where $y_1 = \sigma\sqrt{T} - y_0$ and $y_2 = -y_0$. Thus,

$$C^E(0) = S(0)N(y_1) - Ke^{-rT}N(y_2).$$

This demonstrates that Equation (1) equals Equation (2). Substituting y_0 back into y_1 and y_2, we obtain:

$$y_1 = \frac{\ln(\frac{S(0)}{K}) + T(r + \frac{\sigma^2}{2})}{\sigma\sqrt{T}},$$

and

$$y_2 = \frac{\ln(\frac{S(0)}{K}) + T(r - \frac{\sigma^2}{2})}{\sigma\sqrt{T}},$$

where

$$N(y) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{y} e^{-\frac{x^2}{2}}\,dx.$$

After some mathematical manipulation and substitution, we arrive at the Black-Scholes Formula (1), showing that the option price can be expressed in terms of the cumulative distribution function $N(\cdot)$.

2.2 European Put Options

The Black-Scholes Formula for pricing a European Put option is given by:

$$P^E(0) = Ke^{-rT}N(-y_2) - S(0)N(-y_1) \tag{4}$$

where $P^E(0)$ is the put option price at time $t = 0$, and y_1 and y_2 are defined similarly as for the Call option.

We can derive this formula using Call-Put Parity:

$$S(T) - K = \max\{S(T) - K, 0\} - \max\{K - S(T), 0\} \tag{5}$$

which implies that the difference in the payoffs of a Call and Put option is equal to the difference in their prices.

Applying this formula, we get:

$$e^{-rT}(S(T)-K) = e^{-rT}\max\{S(T)-K,0\} - e^{-rT}\max\{K-S(T),0\}$$
$$e^{-rT}E[S(T)-K] = e^{-rT}E[\max\{S(T)-K,0\}] - e^{-rT}E[\max\{K-S(T),0\}]$$

From the above, we can see that $C^E(0) = e^{-rT}E[\max\{S(T)-K,0\}]$ and $P^E(0) = e^{-rT}E[\max\{K-S(T),0\}]$, so

$$\begin{aligned} e^{-rT}E[S(T)] - e^{-rT}K &= C^E(0) - P^E(0) \\ S(0) - Ke^{-rT} &= C^E(0) - P^E(0) \\ P^E(0) &= C^E(0) - S(0) + Ke^{-rT} \end{aligned}$$

Substituting the Black-Scholes equation for Call options, we get

$$\begin{aligned} P^E(0) &= S(0)N(y_1) - Ke^{-rT}N(y_2) - S(0) + Ke^{-rT} \\ &= S(0)(N(y_1)-1) + Ke^{-rT}(1-N(y_2)) \\ &= Ke^{-rT}N(-y_2) - S(0)N(-y_1) \end{aligned}$$

After applying Call-Put Parity and substituting the Black-Scholes Formula for Call options, we arrive at the Black-Scholes Formula for Put options (4).

In summary, the Black-Scholes Formula provides a theoretical framework for pricing European options based on key parameters such as stock price, strike price, time to maturity, volatility, and risk-free interest rate.

3 Application: Pricing

In this section, we apply the Black-Scholes formula to price European Call and Put options using simulated data. This practical exercise helps demonstrate the application of theoretical option pricing models in real-world scenarios.

3.1 Simulated Data

Consider a hypothetical scenario where we have a stock with a current price of \$100 and an option with a strike price of \$110. The risk-free interest rate is 10% per year, with a volatility of 20%, and the option has a maturity date of 6 months. These parameters are essential inputs for pricing European Call and Put options using the Black-Scholes model.

3.1.1 Calculations By Hand

To compute the option prices manually, we utilize the Black-Scholes formula, which takes into account the stock price ($S(0)$), the strike price (K), risk-free interest rate (r), volatility (σ), and time to maturity (T). The formula for a European Call option is:

$$C^E(0) = S(0)N(y_1) - Ke^{-rT}N(y_2)$$

Where $N(\cdot)$ represents the cumulative distribution function of the standard normal distribution, and y_1 and y_2 are given by:

$$y_1 = \frac{\ln(S(0)/K) + T(r+\frac{\sigma^2}{2})}{\sigma\sqrt{T}}$$
$$y_2 = \frac{\ln(S(0)/K) + T(r-\frac{\sigma^2}{2})}{\sigma\sqrt{T}}$$

Similarly, the formula for a European Put option is obtained by adjusting the Call option formula.

Implementing these formulas we get:

$$y_1 = \frac{ln(\frac{100}{110}) + 0.5(0.1 + \frac{0.2^2}{2})}{0.2\sqrt{0.5}} = -0.249681$$

$$y_2 = \frac{ln(\frac{100}{110}) + 0.5(0.1 - \frac{0.2^2}{2})}{0.2\sqrt{0.5}} = -0.391102$$

Plugging these values into the standard normal distribution we get

$$N(y_1) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{-0.249681} e^{-\frac{x^2}{2}} = 0.401417$$

$$N(y_2) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{-0.391102} e^{-\frac{x^2}{2}} = 0.347861$$

and

$$N(-y_1) = 1 - 0.401417 = 0.598583$$
$$N(-y_2) = 1 - 0.347861 = 0.652139$$

We can now price the Put and Call options,

$$C^E(0) = 100 * 0.401417 - 110 * e^{-0.1*0.5} * 0.347861 = 3.74318 = \$3.74$$
$$P^E(0) = 110 * e^{-0.1*0.5} * 0.652139 - 100 * 0.598583 = 8.37842 = \$8.38$$

3.1.2 Calculations using Python

We implement the Black-Scholes formula in Python to compute option prices. The code calculates the price of both Call and Put options based on the simulated data provided.

```
def Black_Scholes(S, K, T, r, sigma, option_type = 'call'):

  y_1 = (np.log(S/K)+T*(r+(sigma**2/2)))/(sigma*np.sqrt(T))
  y_2 = (np.log(S/K)+T*(r-(sigma**2/2)))/(sigma*np.sqrt(T))

  if option_type == 'call':
    option_price = S*stats.norm.cdf(y_1) - K*np.exp(-r*T)*stats.norm.cdf(y_2)
  elif option_type == 'put':
    option_price = K*np.exp(-r*T)*stats.norm.cdf(-y_2) - S*stats.norm.cdf(-y_1)
  else:
    return "Invalid Option Type"

  return round(option_price,2)
```

```
S = 100     # Current stock price
K = 110    # Strike price
T = (6/12)    # Time to expiration (in years)
r = 0.1     # Risk-free interest rate
sigma = 0.2  # Volatility

print("C^E(0) = ", Black_Scholes(S, K, T, r, sigma, option_type='call'))
print("P^E(0) = ", Black_Scholes(S, K, T, r, sigma, option_type='put'))
```

```
C^E(0) =  3.74
P^E(0) =  8.38
```

Figure 1: Python Code for Option Pricing using Black-Scholes Model

From the output of the Python code, we observe that the price for a European Call option with the given parameters is \$3.74, and for a European Put option, the price is \$8.38. These results demonstrate how the Black-Scholes model can be used to price options based on simulated market conditions.

3.1.3 Payoffs

To better understand the potential outcomes of European Call and Put options, let's examine their payoffs graphically. The payoff graph for the simulated data is depicted below:

Figure 2: Payoff Graph for Simulated Data

This graph visually illustrates the relationship between the option's payoff and the underlying asset's price at expiration.

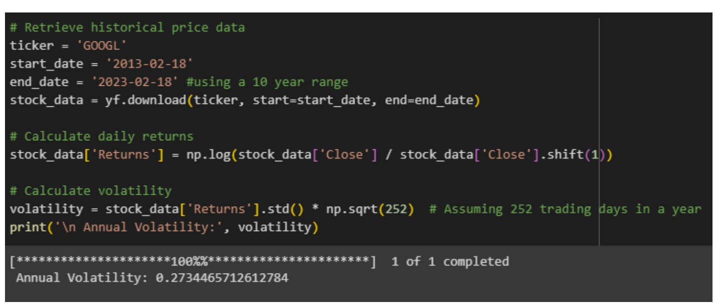
3.2 Real Data

In this section, we transition from simulated data to real-world data by considering options pricing using Google's Alphabet Inc. stock (ticker symbol: GOOGL).

To price European options for GOOGL, we need to gather essential parameters such as the stock price ($S(0)$), strike price (K), risk-free interest rate (r), volatility (σ), and time to maturity (T).

3.2.1 Estimating Volatility

We estimate the volatility for GOOGL by analyzing historical stock prices over the past 10 years. Using Python, we calculate the daily returns and then determine the volatility. The Python code for estimating volatility is provided below:



```
# Retrieve historical price data
ticker = 'GOOGL'
start_date = '2013-02-18'
end_date = '2023-02-18' #using a 10 year range
stock_data = yf.download(ticker, start=start_date, end=end_date)

# Calculate daily returns
stock_data['Returns'] = np.log(stock_data['Close'] / stock_data['Close'].shift(1))

# Calculate volatility
volatility = stock_data['Returns'].std() * np.sqrt(252)  # Assuming 252 trading days in a year
print('\n Annual Volatility:', volatility)

[*********************100%%**********************]  1 of 1 completed
 Annual Volatility: 0.2734465712612784
```

Figure 3: Python Code for Estimating Volatility of GOOGL

Based on the code in Figure 3, the estimated volatility for GOOGL is approximately 27.34%.

3.2.2 Retrieving Data

We retrieve the current stock price ($S(0)$) and risk-free interest rate (r) using Python. The code snippets for obtaining these values are presented below:

```
#Getting todays stock price
stock = yf.Ticker(ticker)
current_price = stock.history(period='1d')['Close'].iloc[-1]

print('Current Stock Price of', ticker, ':', round(current_price,2))

Current Stock Price of GOOGL : 140.52
```

Figure 4: Python Code for Getting Current Price of GOOGL

```
#fred - Federal Reserve Economic Data
#DSG1 - 1 Year Treasury rate
risk_free_rate = web.DataReader('DGS1', 'fred')  # 1-Year Treasury Constant Maturity Rate
current_risk_free_rate = risk_free_rate.iloc[-1]['DGS1'] / 100  # Convert percentage to decimal

print('Current Risk-Free Interest Rate:', current_risk_free_rate)

Current Risk-Free Interest Rate: 0.0493
```

Figure 5: Python Code for Getting Risk-Free Interest Rate

3.2.3 Pricing Options using Python

Using the gathered data, we price European Call and Put options for GOOGL. We set the strike price (K) equal to the current stock price ($S(0)$) and the time to maturity (T) to one year. The Python code for pricing the options is provided below:

```
S = current_price     # Current stock price
K = current_price    # Strike price
T = 1    # Time to expiration (in years)
r = current_risk_free_rate     # Risk-free interest rate
sigma = volatility  # Volatility

print("C^E(0) = ", Black_Scholes(S, K, T, r, sigma, option_type='call'))
print("P^E(0) = ", Black_Scholes(S, K, T, r, sigma, option_type='put'))

C^E(0) =  18.53
P^E(0) =  11.77
```

Figure 6: Python Code for Option Pricing of GOOGL using Black-Scholes Model

Based on the output in Figure 6, the price of the European Call option for GOOGL is \$18.53, and the price for the Put option is \$11.77.

3.3 Pricing Summary

The calculated option prices offer insights into the potential value of the options given market parameters. The higher price of the Put option compared to the Call option suggests a higher expected payoff for the Put option holder in the given scenario. This underscores the influence of factors such as stock price, strike price, and volatility on option prices, as captured by the Black-Scholes model.

Analyzing sensitivity to changes in these parameters can further inform decision-making in financial markets, enabling investors to make more informed choices.

In conclusion, the analysis of real-world options pricing highlights the practical applications of option pricing models and their significance in financial market analysis.

4 Application: Delta Hedging

Delta hedging is a risk management strategy widely used in financial markets to minimize exposure to small changes in the price of an underlying asset. In this section, we apply the Black-Scholes Formula to hedge our portfolio according to its delta value. [CZ11]

4.1 Delta Neutral Portfolio

In delta hedging, we aim to construct a portfolio with a delta value of zero, effectively eliminating exposure to small changes in the underlying asset's price. The portfolio value, $V(S)$, can be expressed as:

$$V(S) = N(y_1)S(0) + y - C^E(0)$$

where: - $N(d1)$ is the cumulative distribution function of the standard normal distribution. - $S(0)$ is the current stock price. - y represents the number of bonds in the portfolio. - $C^E(0)$ is the initial price of the call option.[CZ11]

4.2 Example of Delta Hedging

For our example, we will construct a portfolio consisting of stocks, bonds, and call options. Suppose we wish to write and sell 1000 European call options, we will hedge our portfolio using the same data as above where we have a stock with a current price of \$100 and an option with a strike price of \$110. The risk-free interest rate is 10% per year, with a volatility of 20%, and the option has a maturity date of 6 months.

Our portfolio will consist of three components: stocks (x), bonds (y), and call options (z), where x, y, and z represent the quantities of each asset, respectively.

4.2.1 Calculations by Hand

From the Black-Scholes model, we have determined that the price of a European Call option for the given stock, with a current price of \$100 and a strike price of \$110, is \$3.74.

Given that the calculated delta, $N(y_1)$, is approximately 0.401417 (as shown in Section 3.1.1), we can use this delta value to compute the hedge quantities for our portfolio.

Let's break down the calculations step by step:

1. Option Sale Proceeds: - Selling 1000 European call options at \$3.74 each yields a total of:

$$\text{Option Sale Proceeds} = 3.74 \times -1000 = -\$3740$$

- Therefore, the quantity of call options in our portfolio, z, is -1000.

2. Stock Holdings (x): - The number of stocks we need to hold to hedge the delta is calculated as:

$$x = \Delta \times 1000 = 0.401417 \times 1000 = 401.417$$

3. Bond Holdings (y): - To ensure a delta-neutral portfolio, we need to determine the quantity of bonds, y, required. - Using the formula for the portfolio value $V(S)$, we can solve for y:

$$y = -x \times S(0) - z \times C^E(0)$$

$$y = -401.417 \times 100 - (-1000) \times 3.74 = -36401.70$$

From the calculations above we can see that our portfolio is approximately $x = 401.42$ (stocks), $y = -36401.70$ (bonds), and $z = -1000$ (call options).

Finally, we can verify that our portfolio is indeed delta-neutral by computing its value:

$$x \times S(0) + y + z \times C^E(0) = 401.417 \times 100 + (-36401.70) + (-1000) \times 3.74 = 0$$

Thus, our portfolio's total value is zero, indicating that it is effectively hedged against small changes in the stock price.

4.2.2 Python

We can implement delta hedging programmatically using Python. The code snippet below demonstrates how to hedge the delta of our portfolio using the Black-Scholes model.

```python
def Call_Delta_Hedging(S, K, T, r, sigma, option_quantity):
    #Finding Delta
    y_1 = (np.log(S/K)+T*(r+(sigma**2/2)))/(sigma*np.sqrt(T))
    delta = stats.norm.cdf(y_1)

    # Hedge ratio
    hedge_ratio = delta * option_quantity

    CE =  Black_Scholes(S, K, T, r, sigma, option_type='call')

    #Values for portfolio
    z = option_quantity
    x = -hedge_ratio
    y = -x*S-z*CE

    portfolio = (x,y,z)
    return portfolio
```

```python
S = 100     # Current stock price
K = 110    # Strike price
T = (6/12)    # Time to expiration (in years)
r = 0.1     # Risk-free interest rate
sigma = 0.2  # Volatility

# Option parameters
option_quantity = -1000  # Short (sell) 1000 call options

port = Call_Delta_Hedging(S, K, T, r, sigma, option_quantity)
print(port, "\n")
print(port[0]*S + port[1] + port[2]*Black_Scholes(S, K, T, r, sigma, option_type='call'))
```

```
(401.41715171302985, -36401.71517130299, -1000)

0.0
```

Figure 7: Python Code for Hedging Delta using Black-Scholes Model

The Python code calculates the quantities of stocks, bonds, and call options needed to achieve a delta-neutral portfolio, given the provided parameters.

From the Python code, we can observe that the resulting portfolio quantities are approximately $x = 401.42$ (stocks), $y = -36401.72$ (bonds), and $z = -1000$ (call options). These values confirm that our portfolio is delta-neutral, as intended.

4.3 Hedging Summary

Delta hedging is a powerful risk management technique used by investors to mitigate exposure to changes in the price of an underlying asset. In this section, we applied the Black-Scholes model to construct a delta-neutral portfolio consisting of stocks, bonds, and call options.

Through manual calculations and Python implementation, we demonstrated the process of delta hedging and verified that our portfolio achieves a delta value of zero. By dynamically adjusting the quantities of stocks, bonds, and options, we effectively eliminated sensitivity to small fluctuations in the stock price.

The Python code provided offers a practical approach to implementing delta hedging in real-world scenarios, enabling investors to manage risk and optimize portfolio performance.

Overall, delta hedging provides a valuable strategy for investors to hedge against market volatility and achieve a more stable investment portfolio.

5 Conclusion

In conclusion, the derivation of the Black-Scholes formula and its subsequent applications in option pricing and delta hedging provide valuable insights into the dynamics of financial markets and risk management strategies. By understanding the fundamental principles underlying option pricing, investors can make informed decisions to manage their portfolios effectively.

Through our analysis, we have demonstrated the theoretical framework of the Black-Scholes formula and its practical implementations using simulated and real-world data. The calculated option prices and delta-hedged portfolios illustrate the importance of considering factors such as stock price, strike price, volatility, and time to maturity in financial decision-making.

Furthermore, the Python code snippets presented in this project offer a tangible approach to implementing option pricing models and delta hedging strategies in real-world scenarios. By leveraging computational tools, investors can streamline the process of pricing options and managing risk, thereby enhancing portfolio performance and minimizing exposure to market volatility.

Overall, the Black-Scholes model serves as a cornerstone of modern quantitative finance, providing a robust framework for understanding and pricing derivative securities. By applying the concepts discussed in this project, investors can navigate financial markets with confidence and optimize their investment strategies for long-term success.

In conclusion, the Black-Scholes formula and its applications empower investors to make informed decisions, manage risk effectively, and achieve their financial objectives in dynamic and ever-changing market environments.

References

[CZ11] Marek Capiński and Tomasz Zastawniak. *Mathematics for Finance: An Introduction to Financial Engineering*, chapter 8.3 Black–Scholes Formula; 9.1.1 Delta Hedging, pages 185–190; 192–197. Springer, 2nd edition, 2011.

[Ros23] Sheldon M. Ross. *Introduction to Probability Models*, chapter 10 Brownian Motion and Stationary Processes, pages 653–691. Academic Press, 13th edition, 2023.