Pricing European-Style Bitcoin Call Options Using the Black-Scholes Model

Understanding Options and Black-Scholes Model

An option/options contract is a financial derivative that gives someone the right to purchase an asset at an agreed-upon price at some point in time in the future[1]. An option has a strike price and an expiration date, so in other words, the owner of the option has the "option" to purchase the asset at the strike price sometime before its expiration date. The value of an option is closely related to the volatility of the underlying asset. For example, let's say that option one and option two are contracts on the respective future prices of asset one and asset two, where both options have an expiration date of 30 days and a strike price of $25. If asset one and asset two both possess a 30-day average price of $20, but, over these 30 days, asset one moves between $15 and $25 and asset two moves between $5 and $45, we can deduce that option two was of greater value than option one because at one point in time asset two had the largest difference between its current price and strike price.

Black-Scholes is a partial differential equation used to calculate the price of an option, which means that we can calculate the price of an option using an equation that depends on the partial derivatives of various inputs, such as an estimate of an asset's volatility[2]. This connects back to Bitcoin because Bitcoin is quite volatile, and I wanted to explore whether or not, then why or why not, Black-Scholes can price Bitcoin call options with consistent profit.

[1] "How Options Work for Buyers and Sellers." *Investopedia*, 2 Jan. 2022, www.investopedia.com/terms/o/option.asp

[2] "What Is Black-Scholes Model? Definition of Black-Scholes Model, Black-Scholes Model Meaning." *The Economic Times*, economictimes.indiatimes.com/definition/black-scholes-model

Black-Scholes relies on the following assumptions: transaction costs in the process of purchasing the option are negligible, the volatility of the asset is constant throughout the life of the option, the risk-free interest rate is constant, and dividends are negligible. The Black-Scholes formula (below) returns the price of an option where S denotes the current price of the asset, K denotes the strike price of the option, t denotes time to maturity, and r denotes the risk-free interest rate:

$$C = SN(d_1) - Ke^{-rt}N(d_2)$$

The risk-free interest rate r is included to represent the time value of money as the current inflation rate minus the yield of a Treasury bond matching the time to maturity t. A U.S. Treasury bond is a low-risk alternative investment to a Bitcoin call option and time to maturity refers to the time until the option's exercise date in units of years. r and t are important because they help price the option so that its value is relative to an alternative "risk-free" investment. For example, if a Treasury bond can return 2% in one year, then a $100 investment in a Treasury bond today is more useful than $102 in cash tomorrow because the $100 investment today can be immediately traded as if it already has the value of $102.

σ denotes the standard deviation of the normally or lognormally distributed returns of an asset. Various sources attempt to prove that the prices of assets are lognormally or normally distributed where the mean is the average price of the asset and the standard deviation is a measure of how dispersed each price is from the mean, but, in my opinion, if we understand that for bitcoin especially, the price of a given asset is always left to irrational personal interpretations, it is safest to assume that prices cannot be modeled as a distribution. With that being said, the purpose of this investigation is not to conclude on whether or not distributions of prices are reliable models, so for the sake of data collection, I assumed that this is true and reflected on the effects of this assumption in my conclusion. σ is an important variable because it represents the volatility of an asset as a high value of σ would indicate a small change in price corresponding to a large drop in the probability of occurring, in comparison to a small value of σ.

The function N in Black-Scholes is a cumulative distribution function of a normal distribution, meaning that $N(d_1)$ and $N(d_2)$ are probabilities ranging from 0%-100% expressed as $0 \leq N(d_1) \leq 1$ and $0 \leq N(d_2) \leq 1$. "Standard/Z-scores is a useful statistic which allows us to calculate the probability of a score occurring within our normal distribution and enables us to compare two scores that are from different normal distributions"[3]. $d_1$and d_2 are standardized variables (variables that have a mean of zero and a standard deviation of one) in Black-Scholes so that Black-Scholes is one formula that works for several different requirements. Hence, I defined the function N such that it only requires one input, as the mean and standard deviation can be preset to 0 and 1. This standard normal distribution, a normal distribution of Z-scores, can be calculated manually, but I automated the process using the Python scipy.stats library where $N(d)$ is given by the code snippet: `si.norm.cdf(d, 0.0, 1.0)`.

[3] "Standard Score - Understanding z-Scores and How to Use Them in Calculations." *Laerd Statistics*, statistics.laerd.com/statistical-guides/standard-score.php#:%7E:text=The%20standard%20score%20

The standardized variables d_1 and d_2 are given by the following formulas:

$$d_1 = \frac{ln(\frac{S}{K})+t(r+\frac{\sigma^2}{2})}{\sigma\sqrt{t}} \quad d_2 = \frac{ln(\frac{S}{K})+t(r-\frac{\sigma^2}{2})}{\sigma\sqrt{t}}$$

I then simplified d_2 so that it is an expression in terms of d_1:

$$d_2 = \frac{ln(\frac{S}{K})+t(r+\frac{\sigma^2}{2})}{\sigma\sqrt{t}} - \frac{2t(\frac{\sigma^2}{2})}{\sigma\sqrt{t}} = d_1 - \frac{t\sigma^2}{\sigma\sqrt{t}} = d_1 - \sigma\sqrt{t}$$

We can now deduce that in the first formula $SN(d_1) - Ke^{-rt}N(d_2)$, a simplified way of thinking of $SN(d_1)$ is the product of the current price of the asset and the probability that a buyer will want to exercise the option to purchase the asset, which gives the likely amount received if the option is exercised. Likewise, $Ke^{-rt}N(d_2)$ is the likely payment that will be made when the asset is purchased at the expiration date. Hence, $SN(d_1) - Ke^{-rt}N(d_2)$ returns the price of the option by subtracting what the buyer will likely pay from what the buyer will likely receive.

Data Collection and Data Processing

Bitcoin launched on January 3rd, 2009, so a reasonable range of time for raw data is the last 10 years. Programming and automating at least some of the data collection and processing was necessary to efficiently maintain results that reflect the accuracy and precision of the raw data. I chose to collect data for pricing all possible 10-day, 20-day, 30-day, and 50-day Bitcoin call options as they are the most commonly sold types of options on popular cryptocurrency exchanges. If we consider a 100 day period with 10-day options then the number of prices collected will be 90 values, which I found by taking my range of raw data in units of days subtracted by time to maturity in units of days. Hence, automating data collection was optimal because the number of prices in the data sets I used was 14490:

$$n\, sets \times time - periods\, of\, n\, options = 4 \times 10 \times 365 - 10 - 20 - 30 - 50 = 14490$$

Correspondingly, several variables are required to calculate each value, meaning the number of raw data points is at least in the hundreds of thousands. I began data collection by collecting risk-free rates. This was done using U.S. bonds even though the options are European-style because a follow-up to this investigation would be investigating American-style options if it is determined that Black-Scholes can be used to price Bitcoin call options. r was calculated as[4]:

$$r = \frac{1 + Government\ Bond\ Rate}{1 + Inflation\ Rate} - 1$$

[4] Corporate Finance Institute. "Risk-Free Rate." *Corporate Finance Institute*, 22 Jan. 2022, corporatefinanceinstitute.com/resources/knowledge/finance/risk-free-rate

My range of data collection is 1/3/12 to 12/31/21 (10 years accounting for leap years) because when I was collecting my data the most recent month for Consumer Price Index data from the U.S. Department of Labor Bureau of Labor Statistics was December 2021. Section one of the appendix shows all data used to calculate all risk-free rates. Below is a sample calculation of the risk-free rate on a 10-day option on 12/19/21:

$$r = \frac{1-2.17}{1+0.3} - 1 = \ -1.9\%$$

Volatility is assumed to be constant in Black-Scholes, which is a very problematic assumption that I discuss in my conclusion[5]. Mean is the average of a list of data, so we can find the mean by taking a list of prices separated by the same intervals of time during a day, finding the sum of all the values in that list, then dividing by the total number of values in the list. Values for these daily prices were taken from Investing.com and were evaluated, for credibility, against random values from Yahoo! Finance. Below is a formula showing how I performed calculations for mean:

$$\bar{x} = \frac{\sum_{i=1}^{k} f_i x_i}{n}, \text{ where } n = \sum_{i=1}^{k} f_i$$

[5] "What Is the Black-Scholes Model?" *Investopedia*, 2 Sept. 2021, www.investopedia.com/terms/b/blackscholes.asp#:%7E:text=The%20Black%2DScholes%20Model%20Formula&text=The%20Black%2DScholes%20call%20option,standard%20normal%20probability%20distribution%20function

For a sample calculation we can say that a list (units of US$) comprises a day's open price, high, and low (47123.3, 48553.9, 45693.6) where $n = 3$:

$$\overline{x} = \frac{47123.3+48553.9+45693.6}{3} = US\$\,47123.6$$

"Standard deviation is a measure of how dispersed data is in relation to [its] mean" given by the formula[6]:

$$\sigma^2 = \frac{\sum_{i=1}^{k}(x_i-\overline{x})^2}{n}, \text{ where } n = \sum_{i=1}^{k} f_i$$

The following is a sample calculation for σ using the same list:

$$\sigma = \sqrt{\frac{(47123.3-47123.6)^2+(48553.9-47123.6)^2+(45693.6-47123.6)^2}{3}} = US\$\,1167.71$$

Next, this value then had to be expressed as a percentage of the mean (relative standard deviation) to use it as volatility in my formula for Black-Scholes with my modified inputs:

$$\sigma = \frac{1167.71\times100\%}{47123.6} \approx 2.48\%$$

[6] "Standard Deviation." *National Library of Medicine*, www.nlm.nih.gov/nichsr/stats_tutorial/section2/mod8_sd.html

At this point, the only data I had left to collect was the strike price and current price of the Bitcoin call option. I did this by setting the current price to the average price of the start day and the strike price equal to the product of the current price and what I call a price multiple. My price multiple is simply a value that is incremented linearly and multiplied with the current price to create a vast amount of possible strike prices. For each type of Bitcoin call option that I tested (10-day, 20-day, 30-day, and 50-day) I used my idea of a price multiple, which I set to range from 1-2 incremented 0.1, to collect data on various strike prices that resulted in 44 lines of data on 4 graphs. Before data analysis, below is a sample calculation that prices a 10-day Bitcoin call option with a strike price $k = US\$\,50000$ using values from the previous sample calculations:

$$\sigma = 2.48\%, r = -1.9\%, S = US\$\,47123.6$$

$$d_1 = \frac{ln(\frac{S}{K}) + t(r + \frac{\sigma^2}{2})}{\sigma\sqrt{t}} = \frac{ln(\frac{47123.6}{50000}) + (\frac{10}{365})(\frac{2.48^2}{2} - 1.9)}{2.48\sqrt{\frac{10}{365}}} \approx -0.066$$

$$d_2 = d_1 - \sigma\sqrt{t} \approx -0.066 - 2.48\sqrt{\frac{10}{365}} \approx -0.476$$

$$C = SN(d_1) - Ke^{-rt}N(d_2)$$

$$C \approx (47123.6)N(-0.066) - (50000)e^{1.9(\frac{10}{365})}N(-0.476) \approx US\$\,5632.29$$

Types of these sample calculations were used to test all of my code and below is small portion of my implementation of Black-Scholes in Python:

Figure 1: Option Pricing Function

```python
def price_option(S, K, T, r, sigma):

    # ln(x) is expressed at np.log(x) in NumPy
    d1 = (np.log(S / K) + (r + 0.5 * sigma ** 2) * T) / (sigma * np.sqrt(T))
    d2 = (np.log(S / K) + (r - 0.5 * sigma ** 2) * T) / (sigma * np.sqrt(T))

    call = (S * si.norm.cdf(d1, 0.0, 1.0) - K * np.exp(-r * T) * si.norm.cdf(d2, 0.0, 1.0))

    return call
```

Data Analysis

The figures below depict a graphical representation of profit per day (in units of US$ per day by date) for four types of Bitcoin call options with strike prices at different price multiples every day from 1/3/12 to 12/31/21:

Figure 2: 10-day Bitcoin Call Option

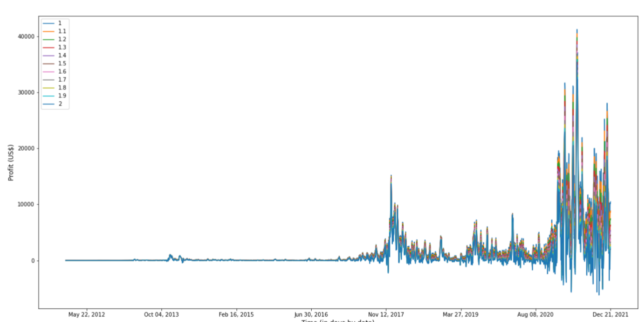


Figure 3: 20-day Bitcoin Call Option

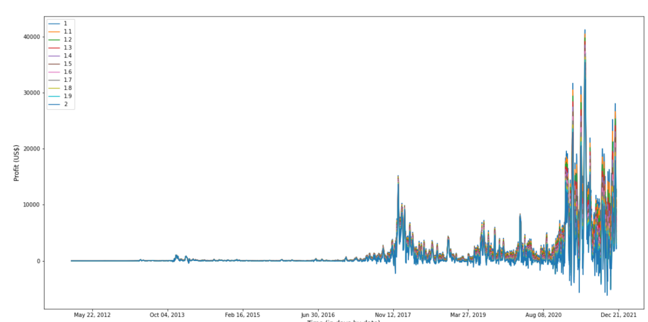


Figure 4: 30-day Bitcoin Call Option

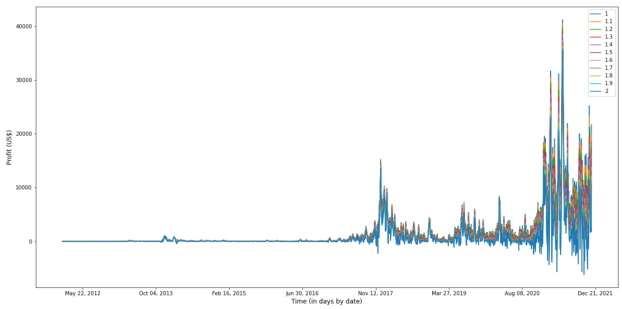
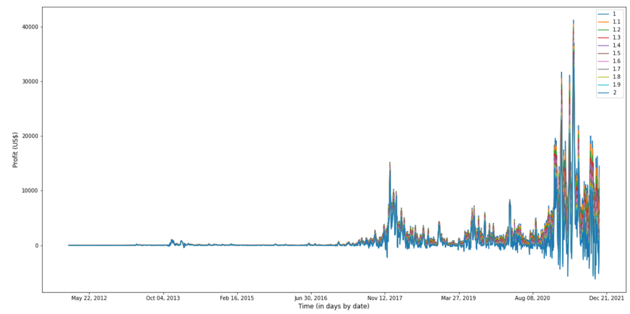
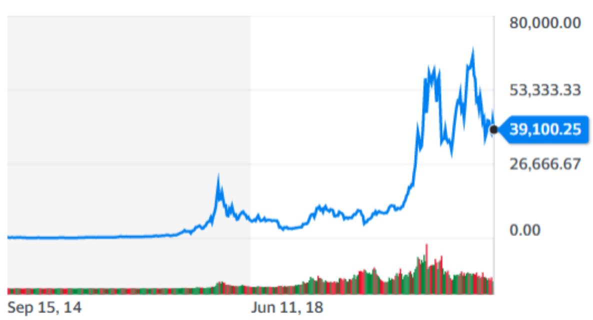


Figure 5: 50-day Bitcoin Call Option



Visually, in Figures 2-5 we can see that each graph appears to have the same shape, which shows that profit from Bitcoin call options may have very similar returns for different times to maturity t, in the extreme short term. As well, it appears as if there was more profit than loss because it seems that the area under the curve above $y = US\$\,0$ could be greater than the area above the curve under $y = US\$\,0$. My last observation was that each graph has components that resemble the shape of the function $f(x) = e^{x}$, which I believe does not come from the term e^{-rt} in Black-Scholes, but rather it is a result of Bitcoin's price movements during the range of my raw data. I corroborated this observation using this BTC-USD graph from Yahoo! Finance[7]:

Figure 6: BTC-USD



[7] BTC-USD graph: https://finance.yahoo.com/quote/BTC-USD?p=BTC-USD&.tsrc=fin-srch

From this, I realized that it was best to express profit and loss as a percentage of the option price C. This is a sample calculation (using $Profit = US\$\,3000$ and $C = US\$\,5000$) for my conversion of option profit expressed as a percentage of the option price:

$$Profit\ (as\ \%\ of\ C) = \frac{Profit \times 100\%}{C} = \frac{3000 \times 100\%}{5000} = \ + 60.0\%$$

After implementing this change I graphed my results:

Figure 7: Profit Per Day (as a percentage of the option price) from 10-day Bitcoin Call Options with Strike Prices as the Price Multiples $1,\ 1.1,\ 1.2,\ 1.3,\ 1.4,\ 1.5,\ 1.6,\ 1.7,\ 1.8,\ 1.9,\ 2$ **of the Current Price Every Day from 1/3/12 to 12/31/21:**

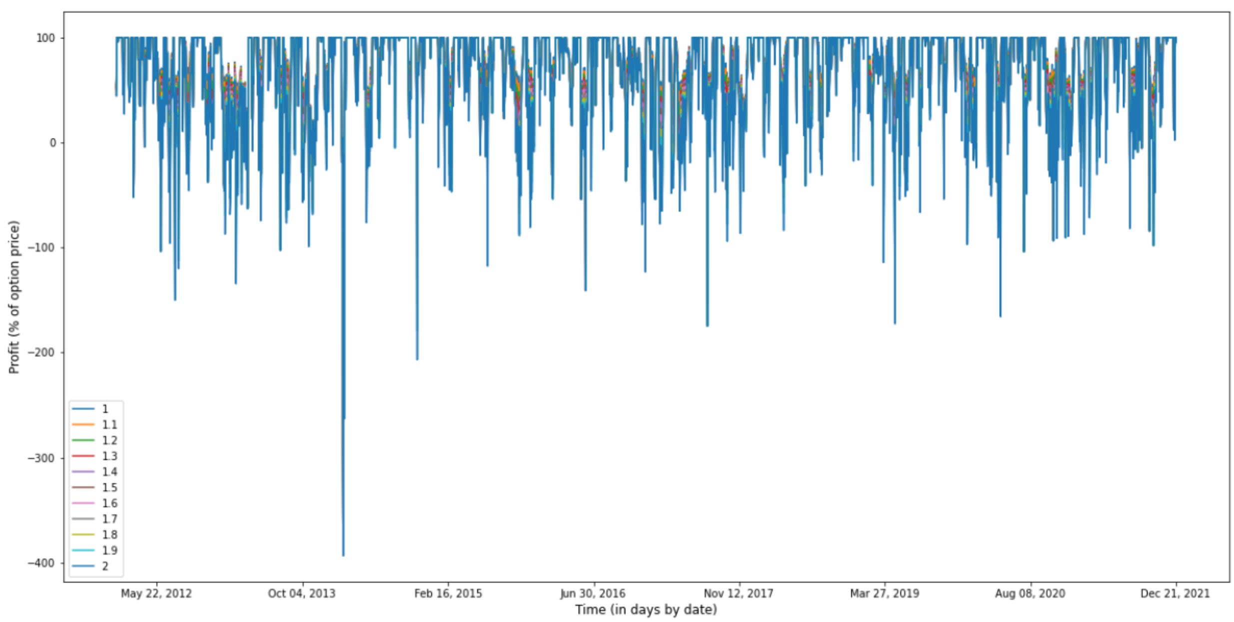


Figure 8: Profit Per Day (as a percentage of the option price) from 20-day Bitcoin Call Options with Strike Prices as the Price Multiples $1,\ 1.1,\ 1.2,\ 1.3,\ 1.4,\ 1.5,\ 1.6,\ 1.7,\ 1.8,\ 1.9,\ 2$ **of the Current Price Every Day from 1/3/12 to 12/31/21:**

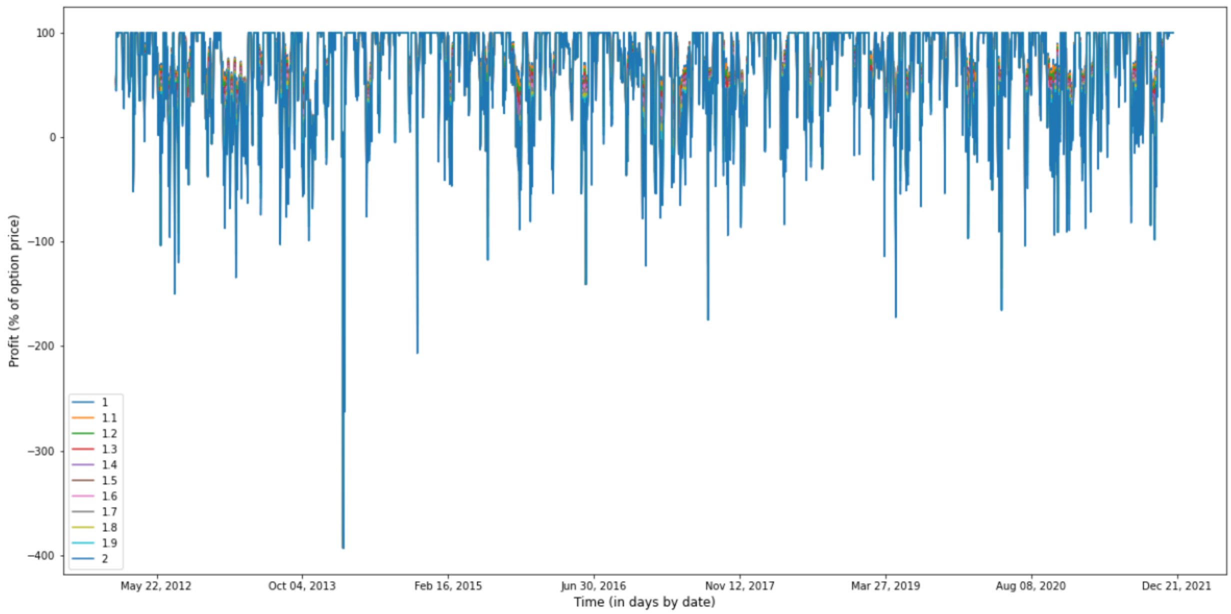


Figure 9: Profit Per Day (as a percentage of the option price) from 30-day Bitcoin Call Options with Strike Prices as the Price Multiples $1,\ 1.1,\ 1.2,\ 1.3,\ 1.4,\ 1.5,\ 1.6,\ 1.7,\ 1.8,\ 1.9,\ 2$ **of the Current Price Every Day from 1/3/12 to 12/31/21:**

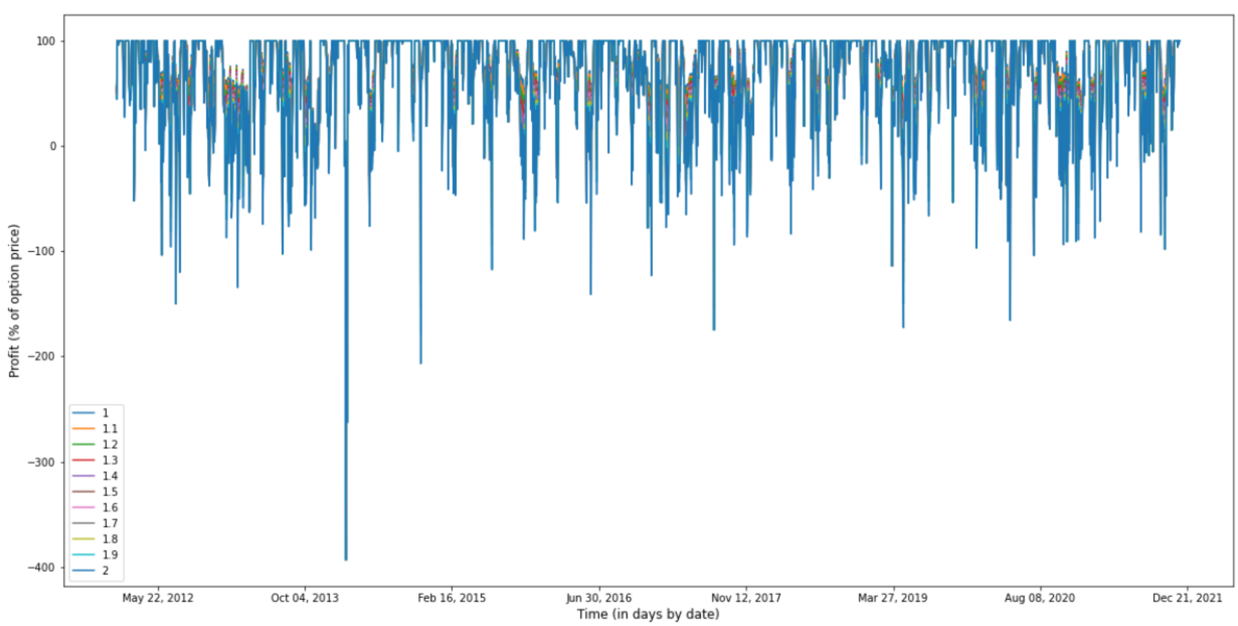
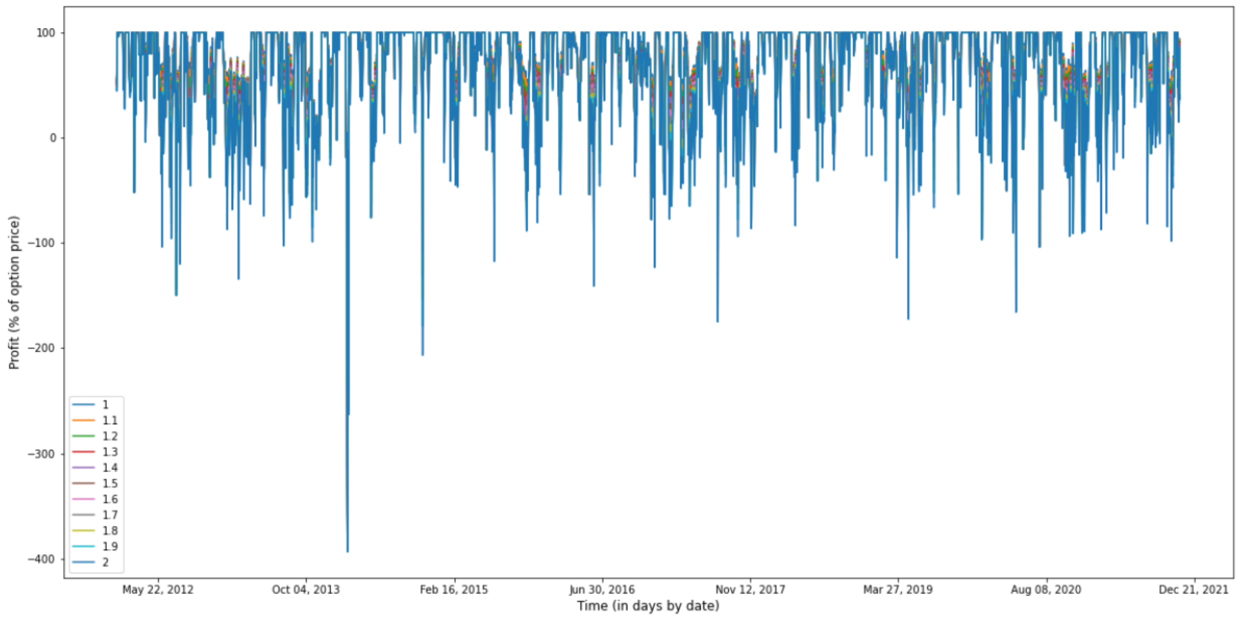


Figure 10: Profit Per Day (as a percentage of the option price) from 50-day Bitcoin Call Options with Strike Prices as the Price Multiples $1, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2$ **of the Current Price Every Day from 1/3/12 to 12/31/21:**



To interpret Figures 7-10 we must understand that they do not have the same units as a graph of an asset, such as a stock, where the y-axis would be price and the x-axis would be time. In Figures 7-10 the y-axis is the percent return of Bitcoin call options sold that day at their respective strike prices, and the x-axis is time. Hence, unlike a graph for a stock, the percent return is not seen as the percent change in price for a given period of time on a graph. Instead, it is the difference between profit and loss, which is the difference between the area under the curve above $y = 0\%$ and the area above the curve under $y = 0\%$, for a given period of time. Likewise, this is why Figures 7-10 show us that maximum percent return can only ever be 100%, which occurs when $C = Profit$. To complete data analysis and determine whether or not my implementation of Black-Scholes can be used to price Bitcoin call options, I took the (definite) integral of two types of each curve to yield the difference between the area under the curve above $y = 0\%$ and the area above the curve under $y = 0\%$ as the total percent return for each type of option. I accomplished this using the Matplotlib library for Python and I did not include a sample calculation due to the complexity of the calculation.

Figure 11: Total Percent Return (%) for 10-day, 20-day, 30-day, and 50-day Bitcoin Call Options from 1/3/12 to 12/31/21 and Their Corresponding Price Multiples

Price Multiple	10-day	20-day	30-day	50-day
1	15255%	15190%	15127%	15009%
1.1	14807%	14744%	14681%	14564%
1.2	14385%	14324%	14261%	14145%
1.3	14003%	13943%	13880%	13765%
1.4	13653%	13595%	13532%	13418%
1.5	13342%	13285%	13222%	13108%
1.6	13065%	13010%	12947%	12833%
1.7	12811%	12756%	12692%	12579%
1.8	12573%	12518%	12454%	12341%
1.9	12350%	12295%	12232%	12119%
2	12150%	12095%	12032%	11919%

Figure 11 shows the total percent return where every percent return is above 10000%, which proves that my initial observation of there appearing to be more profit than loss in Figures 2-5 was true. Also, the total percent return appears to be inversely proportional to my price multiple, or in other words, the total percent return appears to be inversely proportional to the change in strike price. If every variable aside from the strike price K in $C = SN(d_1) - Ke^{-rt}N(d_2)$ is held constant, then C will decrease as K

increases, meaning percent return decreases as well, causing the inverse proportionality. Ergo, from Figure 11, I can come to the conclusion that based upon my raw data, selling short-term European-style Bitcoin call options from 1/3/12 to 12/31/21 would have been immensely profitable. However, there are many assumptions made in Black-Scholes and possible errors/bias in my raw data that have skewed results in making this conclusion.

Conclusion

To evaluate the problematic assumptions in this investigation we can look at the inputs to the Black-Scholes model ($S,\ K,\ t,\ r,\ \sigma$) individually. The current price S was taken as the average price of Bitcoin on the starting day of each option, and since this value comes from a reputable source, it is not a source of error. Likewise, the time to maturity t and the strike price K are not sources of error within the raw data for the same reason. r seems as if it would be a source of error because it was calculated using the U.S. inflation rate, which is based on vast amounts of inaccurate data. However, a buyer considering U.S. Treasury bonds as a low-risk alternative investment would base their decision on the same data with the same errors, rendering the error in pricing the option obsolete. In my opinion, the biggest source of error in this investigation was volatility σ. As stated previously, σ is a value calculated on the assumption that Bitcoin's daily price movements are normally or lognormally distributed. This does not create a systematic error in my data, but it slightly randomizes the option's prices, making them less optimal. Also, the raw data used to find σ consisted of a list of daily prices separated by equal periods of time during the day, but, still on the assumption that σ is a useful value, a considerably more accurate approach would be using data that comprises every single daily price for a given day. The final problematic assumption is that volatility σ is constant during the life of the option.

Figure 12: Bitcoin Volatility Index[8]

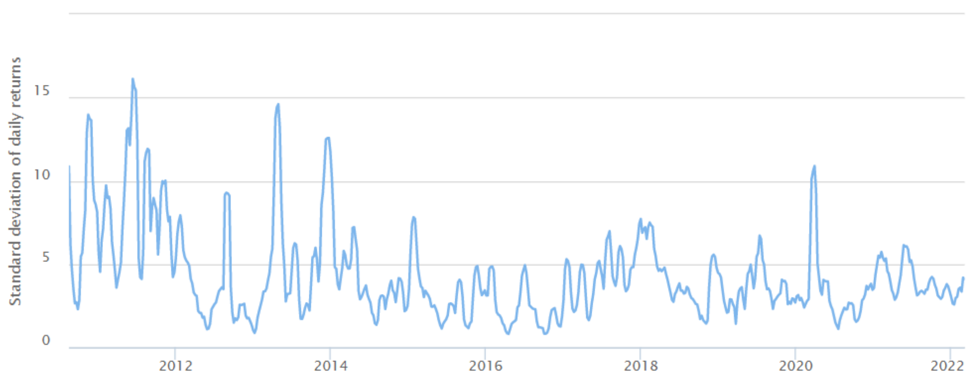


Figure 12, provided by Buy Bitcoin Worldwide, is a graph that displays Bitcoin's daily change in volatility over time, which clearly shows that the assumption that σ is constant, even in the short-term, is incorrect. Hence, σ causes multiple types of random error present in my results in Figure 11. To mitigate both errors caused by the assumptions that prices follow a distribution and that σ is constant, I could have backtested σ with historical average results to create values for what the maximum and minimum prices of each Bitcoin call option should be to then filter out types of options whose range of maximum and minimum prices are deemed too large. To accomplish modeling the randomization of σ, the best way probably would have been to use a Taylor series, which is a method of approximating a function as a polynomial with an infinite number of terms using the function's derivatives at each point. These seem to be the only errors present in my raw data because automating my data collection allowed

---

[8] Bitcoin Volatility Index graph: https://www.buybitcoinworldwide.com/volatility-index/

me to eliminate random errors by using a large amount of raw data and random errors created by rounding. Nonetheless, even though many errors were eliminated, this implementation of Black-Scholes to price short-term European-style Bitcoin call options may not be ready for real-world use.

Something important to address in my raw data is that the Bitcoin call options seem overpriced. For example, take a Bitcoin call option that has a current price of $S = US\$\,50000$, a strike price of $K = US\$\,51000$, a price of $C = US\$\,7000$, and a buyer shorting Bitcoin because they have reason to believe that Bitcoin will fall to a significantly lower value. The buyer would purchase this option as "insurance" on this bet to cover a portion of their losses in the worst-case scenario. That being said, the reason why my implementation of Black-Scholes is not yet ready for real-world use is that my results are for European-style options, not American-style options.

European-style options are options that can only be exercised at the exact end of the time to maturity t, whereas American-style options are options that can be exercised at any point in time during the life of the option. To price Bitcoin call options in the real world on an exchange, the industry-standard would be for them to be American-style. Investopedia states that Black-Scholes should be avoided for American-style options and that there are alternatives such as the Binomial Option Pricing Model[9]. However, for real-world Bitcoin call option pricing, it would be best practices to first test my implementation by modifying return to equal $C - highest\ \Delta Bitcoin\ price\ during\ t$ to get

[9] "Circumventing the Limitations of Black-Scholes." *Investopedia*, 30 Jan. 2022, www.investopedia.com/articles/active-trading/041015/how-circumvent-limitations-blackscholes-model.asp#:%7E:text=The%20Black%2DScholes%20model%20does%20not%20account%20for%20the%20early,as%20the%20Binomial%20pricing%20model

all lowest possible returns to determine if the model still holds up. Lastly, similar to how historical prices are no indicator of future prices, the continued use of this model to price future Bitcoin call options is left to interpretation, and it is important to keep in mind that no quantitative model should be followed blindly as that would pose a great deal of risk.

Works Cited

"How Options Work for Buyers and Sellers." *Investopedia*, 2 Jan. 2022, www.investopedia.com/terms/o/option.asp

"What Is Black-Scholes Model? Definition of Black-Scholes Model, Black-Scholes Model Meaning." *The Economic Times*, economictimes.indiatimes.com/definition/black-scholes-model

"Standard Score - Understanding z-Scores and How to Use Them in Calculations." *Laerd Statistics*, statistics.laerd.com/statistical-guides/standard-score.php#:%7E:text=The%20standard%20score%20

Corporate Finance Institute. "Risk-Free Rate." *Corporate Finance Institute*, 22 Jan. 2022, corporatefinanceinstitute.com/resources/knowledge/finance/risk-free-rate

"What Is the Black-Scholes Model?" *Investopedia*, 2 Sept. 2021, www.investopedia.com/terms/b/blackscholes.asp#:%7E:text=The%20Black%2DScholes%20Model%20Formula&text=The%20Black%2DScholes%20call%20option,standard%20normal%20probability%20distribution%20function

"Standard Deviation." *National Library of Medicine*, www.nlm.nih.gov/nichsr/stats_tutorial/section2/mod8_sd.html

"Circumventing the Limitations of Black-Scholes." *Investopedia*, 30 Jan. 2022, www.investopedia.com/articles/active-trading/041015/how-circumvent-limitations-blackscholes-model.asp#:%7E:text=The%20Black%2DScholes%20model%20does%20not%20account%20for%20the%20early,as%20the%20Binomial%20pricing%20model