

Real-Time Options Pricing Model

Jaisal Friedman

December 16 2018

Abstract

This paper explores the Black-Scholes-Merton options pricing model, derives a predictive extension model, and visualizes both models in comparison to real-time pricing options pricing. The paper also explores various methodologies of calculating historical volatility. A portfolio of 5 U.S. Market Stocks and 1 index fund was taken as example for the project. The model was limited to visual analysis from real-time simulations as further explained in the extensions section.

1 Introduction

The impact of the Black-Scholes-Merton model on modern financial markets drove the paper's initial fascination. The paper intends to visually model the Black-Scholes-Merton Model, a predictive extension model, and the real-time options spread for a better understanding of the options market. This project first seeks to explore the dynamics of the Black-Scholes-Merton and its derivation. The paper then introduces a predictive extension to which a visual analysis is displayed in real-time for one to decipher its effectiveness. Four methodologies for the historical volatility calculation were explored and compared in real-time between the 6 example assets. The Black-Scholes-Merton calculation for real-time assets with historical data was then modeled. The extended predictive model was derived for real-time assets. And lastly, the F-B-S, predictive model, and real-time options pricing was visually displayed and analyzed using 3D volatility surfaces. The paper, itself, will begin with the methodology: where the historical volatility calculations, assumptions of the Black-Scholes-Merton model, and the predictive extension will be discussed. Then introduce the implementation: where the real-time numerical simulation, 3D visualization, and discussion of results are presented. Lastly, it will conclude with a short conclusion, explanation of the software development, limitations, and possible extensions and improvements of the model.

2 Methodology

2.1 Introduction to the Black-Scholes-Merton Model

The Black-Scholes model was initially published in 1973. It provided the first deterministic pricing model for the options market. This led to a boom in the options financial markets that has continued till present day. In 1976, Merton published his extension of the Black-Scholes model which included the effect of dividend payouts. Today, the Black-Scholes-Merton model is used to determine the Greeks and Implied Volatility in the financial options market. Its impact on efficiently pricing the options market as a benchmark model holds strong even some 40 years later. For this reason, this paper starts its extension and comparative visualization with such a model.

2.2 Black-Scholes-Merton Equation

$$C_t(s, t) = S \cdot e^{-qt} \cdot N(d_1) - X \cdot e^{-rt} \cdot N(d_2)$$

$$P_t(S, t) = -e^{-qt} \cdot S \cdot N(-d_1) + X \cdot e^{-rt} \cdot N(-d_2)$$

where:

$$d_1 = \frac{\ln(\frac{S}{X}) + (r - q + \frac{\sigma^2}{2}) \cdot t}{\sigma \cdot \sqrt{t}}$$

$$d_2 = d_1 - \sigma \cdot \sqrt{t}$$

$$N(x) = \int_{-\infty}^x \frac{1}{\sqrt{2\pi} \cdot \sigma^2} \cdot e^{-\frac{(u-\mu)^2}{2\sigma^2}} du, \text{ where } \sigma^2 = 1, \mu = 0$$

2.2.1 Notable Assumptions

1. The cumulative distribution function $N(x)$ follows market probabilities. In fact, there are unpredictable Black Swan events which severely skew market movements. Fat Tail distributions are more accurate for financial markets.
2. Market speculation has no effect on the pricing of options.
3. The option is a European option meaning it can only be exercised at expiry. This makes the mathematics infinity more simple. However, for American options this is false; they can be executed at any moment.

2.2.2 Historical Volatility σ

Historical volatility in the Black-Scholes-Merton σ can be calculated by different methods. Classically, σ is calculated by either the standard deviation of the log returns or the percent change of close-to-close asset pricing for the past 30 days normalized to the last trading year. This methodology utilizes

the close-to-close historical pricing. The log returns is argued to be a more mathematically sound indicator than percent change for σ .

$$hv_{percent} = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2}, \text{ where } x = \frac{s(t)_{close} - s(t-1)_{close}}{s(t-1)_{close}}$$

$$hv_{log} = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2}, \text{ where } x = \log \frac{S_{close}^t}{S_{close}^{t-1}}$$

However, Historical volatility in the Black-Scholes-Merton σ can also be calculated by intraday methods. This paper presents two possible extensions. The first is self-derived. It utilizes both the open-close and high-low pricing to return a weighted intraday historical volatility. The second is the highly referenced Parkinson Historical Intraday volatility calculation. Both methodologies utilizes the intraday historical pricing and provide a different outlook at on σ .

$$hv_{wext} = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2}, \text{ where } x = \log \frac{S_{low}^{t_{high}}}{S_{low}^{t-1}} \cdot w + \log \frac{S_{low}^{t_{high}}}{S_{low}^{t-1}} \cdot (1 - w)$$

$$hv_{park} = \sqrt{\frac{1}{N} \sum_{i=1}^N \frac{1}{4 \ln(2)} (x_i)^2}, \text{ where } x = \log \frac{S_{low}^{t_{high}}}{S_{low}^{t-1}}.$$

2.3 Predictive Extension

The predictive extension presented in this paper builds off the Black-Scholes-Merton model. The model attempted to scale the B-S-M model by sigmoid function encapsulating current financial market health and beta exposure of the underlying asset to the market conditions.

2.3.1 Equation

$$M(c_1, c_2, dd, \beta, t) = \frac{x}{1 + (|x|)} + 1, \text{ where } x = \frac{c_1 \beta \sqrt{t}}{dd} - c_2$$

$$C_{ext}(s, t) = S \cdot e^{-qt} \cdot N(d_1) - X \cdot e^{-rt} \cdot N(d_2) \cdot M(c_1, c_2, dd, \beta, t)$$

$$P_{ext}(S, t) = (-e^{-qt} \cdot S \cdot N(-d_1) + X \cdot e^{-rt} \cdot N(-d_2)) \cdot \frac{1}{M(c_1, c_2, dd, \beta, t)}$$

where $c_1, c_2 = \text{constants}$, $dd = \text{defaulted debt}$ and $\beta = \text{risk exposure of an asset to the market}$

2.3.2 Notable Assumptions

1. Speculation about the current market conditions has an impact on the pricing of long-dated options.
2. The β exposure of an underlying asset to the market affects the pricing of certain assets over assets with a lower β .

3. The time of the options has a square-root proportionality with the pricing of options under certain market conditions. Longer-dated options will be square-root proportionally more affected by overall market speculation.
4. The dd, defaulted debt, indicator can be effectively represent the overall financial market health.
5. The M function follows a Sigmoid proportionality with the Black-Scholes pricing model centered by c_1, c_2 .

2.4 Real-Time Pricing

The real-time pricing involved a substantial amount of software development. This was deemed worthy as the model's visual analysis is better implemented in real-time.

2.4.1 Assets Chosen

- MSFT - Microsoft
- BA - Boeing
- V - Visa
- AAPL - Apple
- GOOGL - Alphabet Class A
- SPY - SPDR S&P 500 ETF Trust

These assets were chosen directly from my personal portfolio as they held the most interest to me. In the code, one can select their own portfolio assets to visualize.

2.4.2 API Interactions

1. FRED - Federal Reserve Economic Data
2. IEX Exchange - Real-time Stock Data
3. Tradier - Real-time Options Data
4. Plotly - Interactive 3D Visualizations

3 Implementation

3.1 Historical Volatility σ

The implementation of the methodology was carried out in python. A working model was built to take real-time data and generate the appropriate visualization.

```
def hv_log(historicals, portfolio):
    dates = historicals['date']
    df = {}
    for stock in portfolio:
        sigmas = {}
        h = historicals[stock].join(dates)
        i = timedelta(days = 31)
        j = 1
        while(i+dates[j] <= dates[len(dates)-1]):
            h1 = h[(h['date'] >= dates[j-1]) & (h['date'] <= i+dates[j])]
            h1['volatility'] = h1['close'].pct_change()
            h1[1:]
            sigma_matrix = np.std(h1['volatility'])* np.sqrt(252)
            sigmas[i+dates[j]] = sigma_matrix
            j+=1
        df[stock] = sigmas
    return pd.DataFrame(df)
```

3.2 Strike Prices

Strike prices were calculated for each stock instead of scraped.

```
def strikes_(s, num):
    strikes = {}
    for ticker in s:
        if (s[ticker]>300):
            strikes[ticker] = [((round(s[ticker],-1))+((i-num)*10)) for i in range(2*num)]
        else:
            strikes[ticker] = [((round(s[ticker],-1))+((i-num)*5)) for i in range(2*num)]
    return strikes
```

3.3 Expiration

Expiration was also calculated for each stock. This was tricky because options expire on the 3rd Friday of each quarter. Because the project focused on longer-dated options. The code generated expiries from the current date for the next 3 quarters, and then 2 half-years from then. Although the code works efficiently, not all stocks follow these dates. There were 3 exceptions with the case examples which the Tradier library caught later. These had to manually fixed in Jupyter.

```

def third_friday(month, year):
    monthcal = c.monthdatescalendar(year, month)
    return monthcal[2][-1]
def generate_times(start, year, options_dates):
    t = []
    for i in range(6):
        if (i < 3):
            t.append(third_friday(options_dates[(start+i)%len(options_dates)], (year + int(
            else:
            t.append(third_friday(options_dates[(start+i+(i-2))%len(options_dates)], (year +
    return t
def t_(today, options_dates):
    t = []
    month = int(today.month-1 + today.day/third_friday(today.month, today.year).day) % 12
    if (month > 9):
        t = generate_times(0, today.year+1, options_dates)
    elif (month > 6):
        t = generate_times(3, today.year, options_dates)
    elif (month > 3):
        t = generate_times(2, today.year, options_dates)
    else:
        t = generate_times(1, today.year, options_dates)
    return t

```

3.4 Options Pricing

These are the options pricing functions that were used in Jupyter to calculate real-time pricing of each model.

```

def M_(c1, c2,default, beta, t):
    x = c1*beta*np.sqrt(t)/default - c2
    return x/(1+abs(x))+1
def call_(s, x, t, r, q, d1, d2):
    return s*math.exp(-q*t)*norm.cdf(d1)-x*math.exp(-r*t)*norm.cdf(d2)
def put_(s, x, t, r, q, d1, d2):
    return x*math.exp(-r*t)*(1-norm.cdf(d2)) - s*math.exp(-q*t)*(1-norm.cdf(d1))
def d1_(s, x, t, r, q, sigma):
    return (np.log(s/x)+(r-q+(np.power(sigma,2)/2)*t)) / (sigma*np.sqrt(t))
def d2_(d1, sigma, t):
    return d1 - sigma*np.sqrt(t)

```

3.5 Model Extension Indicators

The US High Yield Master 2 Option-Adjusted Spread was used for the defaulted rate in the extended model. This is an economic indicator from FRED, which tracks the percent of high-yield bond defaults in the US. See

figure 7 for reference.

This was used for the following reasons:

1. It has historically predated US financial market crashes. Check 2007 and 2001.
2. It represents the most leveraged defaulted debt within the financial markets this paper is addressing.

The β values were pulled from the IEXFinance API. β is the historical risk of an asset to the market fluctuations. It can be calculated by a regression analysis between historical asset and market returns. The β for the SPY should be close to 1, while more volatile stocks like GOOGL will be closer to 1.5.

3.6 Results

1. Historical Volatility

The figures section contains the plots for each historical volatility calculation. There are two plots which show all 4 calculations on one. This was done for BA and AAPL as examples.

2. 3D Volatility Surfances The volatility surfaces for each stock and each call/put are displayed below in the Figures section. However, there are interactive figures available to explore. The link is referenced in the Appendix. See Figure 8-18 in Appendix and 3D Surfaces in Appendix under Code Base.

These can be seen at in the Appendix section under Figures

4 Conclusion

4.1 Discussion

- **Historical Volatility**

Historical Volatility was presented by 4 different methodologies. After the visualization of these 4 methodologies; it became clear that the classical calculation hv_{log} made the most sense for the longer-term options I was considering. The intraday calculations, although interesting, often hold volatility that the market would not consider influential in longer dated options.

- **Black-Scholes-Merton Extension**

The Black-Scholes-Merton Extension as seen in the figures below often intersects the Black-Scholes-Merton model surface and the Real-time Spread surface. More so, the Extension favors time as a more defining parameter than the Black-Scholes-Merton or Spread models. It is impossible to evaluate the effectiveness of the extension model on predicatively pricing future options. Because the predictive model implies that the current market conditions will effect future prices of the options and thus both the current Spread and the Black-Scholes-Merton model may not reflect the true options pricing. This is further discussed in the Extensions section below.

- **3D Volatility Surfaces**

The 3D Volatility Surfaces are a great tool to explore market inefficiencies in options pricing. As a definitive analysis it is difficult to evaluate the effectiveness of the models in pricing options without taking into affect the change in options prices between days. This is because the market may contain arbitrary inefficiencies on any given day.

4.2 Improvements

- **3D Surfaces for each Historical Volatility**

This would be an effective and interesting improvement to visualize the effect of σ calculations on the 3D volatility surfaces.

- **Parameter sweep for c_1, c_2**

A Parameter sweep over c_1, c_2 would be a better measure to fit the extension to current real-time market spread pricing. This would need to be done over-time to ensure that certain market conditions at a singular point in time don't dominate the sweep.

4.3 Extensions

- **Fixing the Improvements**

Fixing the listed Improvements in the above section would clarify a more

applicable 3D volatility surface and thus model extension. This could be used to further evaluate the effectiveness of the model extension

- **Back-Test**

A back-test to validate the model would be a key extension. There are several ways this could take place. For example, the extension could simply track the pricing each day of all three-models and present a visualization of how each fluctuates with the market. Or the back-test could trace a period of high-default like 2007 to understand how the model would price options and if this would be effective in the long-run. One could also take the approach of tracking pricing of the models when it would be purchased vs. when it would expire. This would need some more distilled thought, however.

- **American Binary Options**

The Black-Scholes-Merton equation and thus its extension implies that the option will only be exercised at expiry. This is untrue of American options and thus could be optimized by using a Binary American options pricing model.

- **Cumulative Distribution Function**

The Cumulative Normal Distribution Function which the Black-Scholes-Merton uses to imply probabilities of future pricing changes on the underlying asset may not be an optimal technique to model the financial markets. One could try a Multi-Fractal approach or a Fat-tails distribution which more heavily weights highly improbable events and lessens the likelihood of slightly improbable events. This has been shown to more accurately model probabilities of pricing changes in financial market. This could be included in the back-test to use historical market conditions as a benchmark for accurateness of the model extension.

5 Appendix

5.1 References

5.1.1 Bibliography

- Black, Fischer and Scholes, Myron, (1972), The Valuation of Option Contracts and a Test of Market Efficiency, Journal of Finance, 27, issue 2, p. 399-417, <https://EconPapers.repec.org/RePEc:bla:jfinan:v:27:y:1972:i:2:p:399-417>.
- Black, Fischer. "The pricing of commodity contracts." Journal of financial economics 3.1-2 (1976): 167-179.

- Spurgin, Richard B., and Thomas Schneeweis. "Efficient Estimation of Intra-day Volatility: A Method of Moments approach incorporating Trading Range." Financial Markets Tick by Tick (1999).
- Taleb, Nassim. Dynamic hedging: managing vanilla and exotic options. Vol. 64. John Wiley Sons, 1997.

5.1.2 API References

- FRED API - <https://research.stlouisfed.org/docs/api/fred/>
- Tradier API - <https://developer.tradier.com/>
- IEX Exchange API - <https://iextrading.com/developer/docs/#getting-started>
- Plotly API - <https://plot.ly/python/>

5.2 Code Base

The project was written in python.

- Github - https://github.com/jaisal1024/options_realtime_modeling
- Interactive Volatility Surfaces - <https://plot.ly/~jaisal1024/#/>

5.3 Figures

For the Volatility Surface the key is as follow:

- Red = Black-Scholes-Merton Predictive Extension
- Blue = Black-Scholes-Merton Model
- Green = Real-time Spread

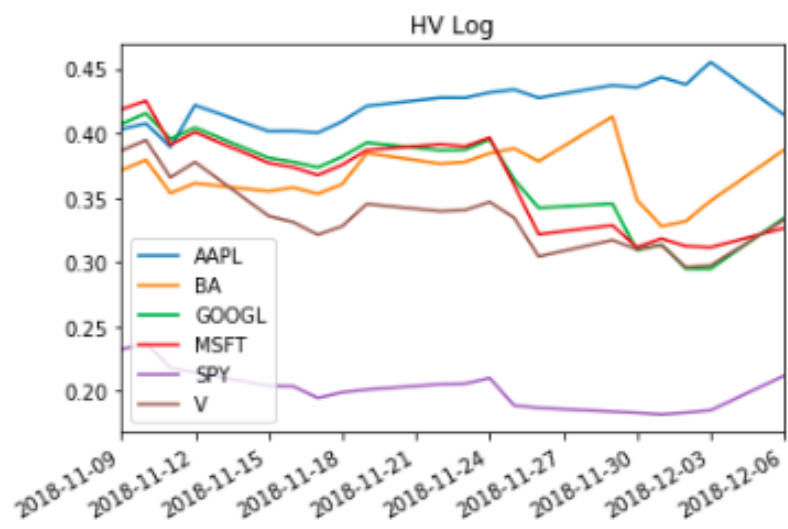


Figure 1: Historical Volatility - Log

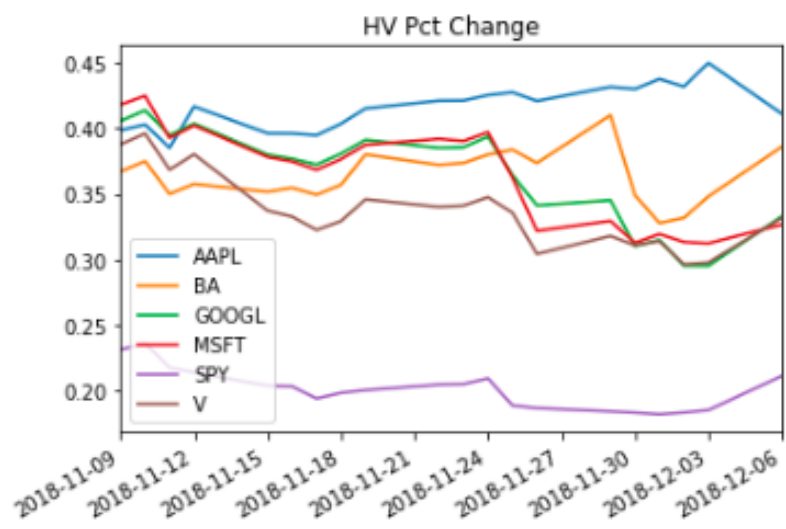


Figure 2: Historical Volatility - Pct

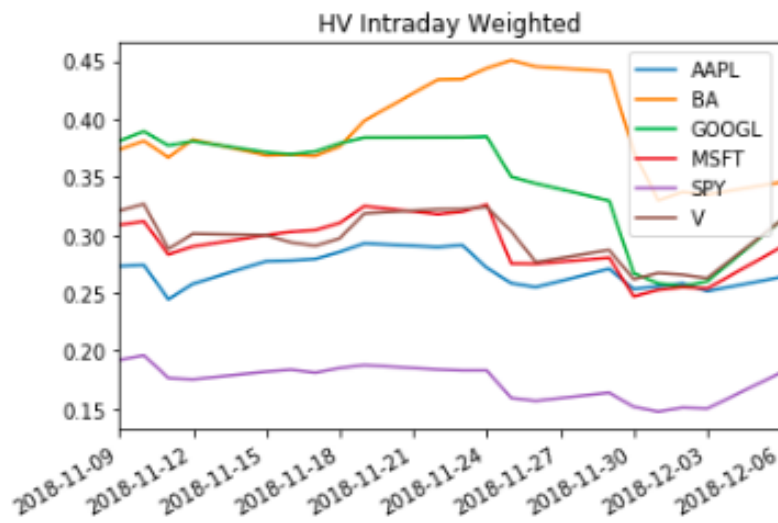


Figure 3: Historical Volatility - Intraday Weighted

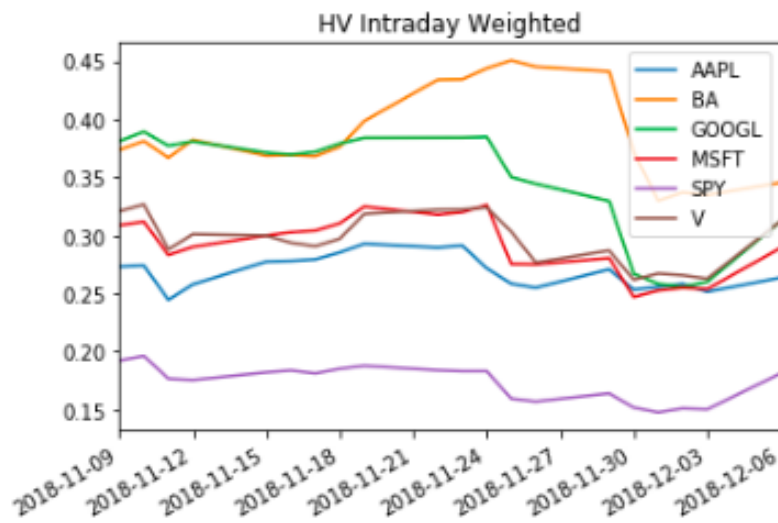


Figure 4: Historical Volatility - Intraday Parkinson

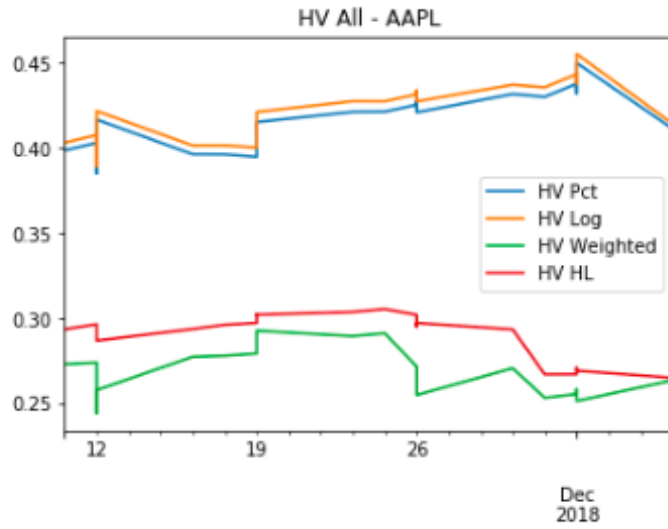


Figure 5: Historical Volatility - All AAPL

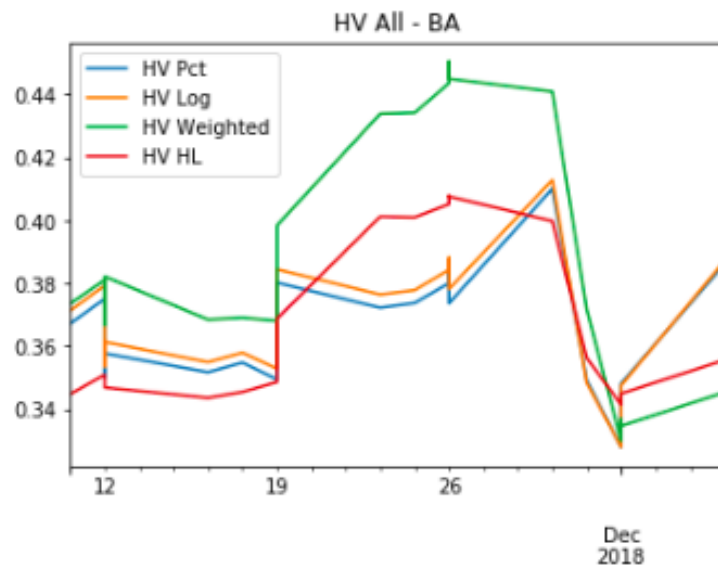


Figure 6: Historical Volatility - All BA

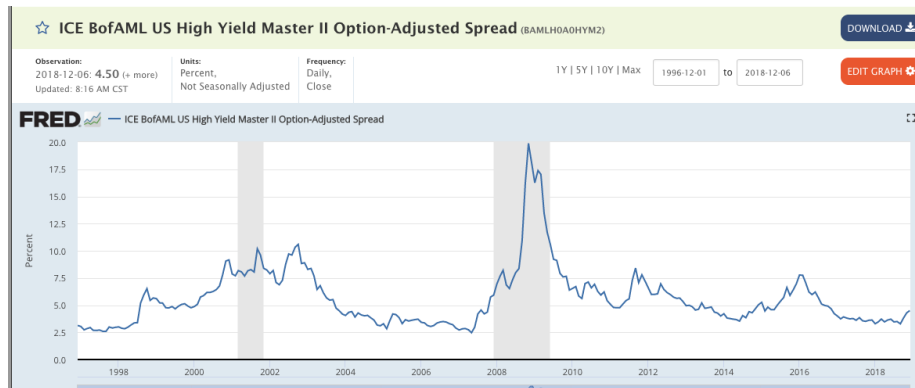


Figure 7: FRED Defaulted Debt Indicator

AAPL Call Volatility Surface

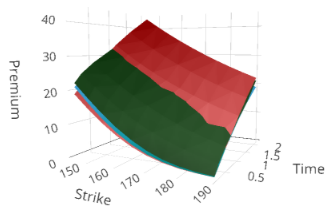


Figure 8: Apple Call Volatility Surface

AAPL Put Volatility Surface

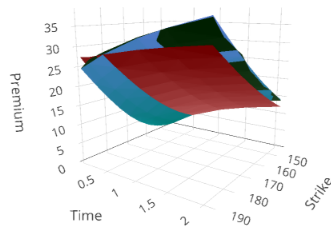


Figure 9: Apple Put Volatility Surface

BA Call Volatility Surface

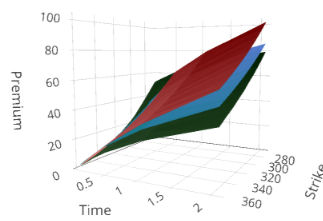


Figure 10: Boeing Call Volatility Surface

BA Put Volatility Surface

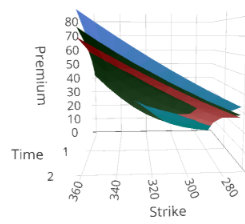


Figure 11: Boeing Put Volatility Surface

GOOGL Call Volatility Surface

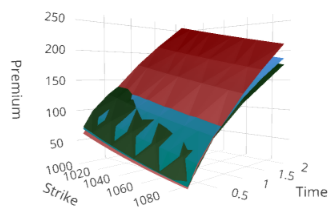


Figure 12: Google Call Volatility Surface

GOOGL Put Volatility Surface

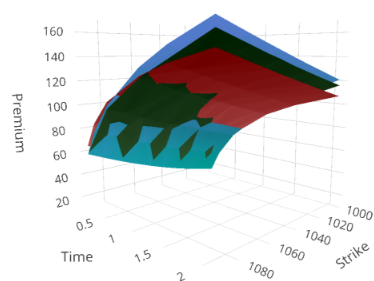


Figure 13: Google Put Volatility Surface

MSFT Call Volatility Surface

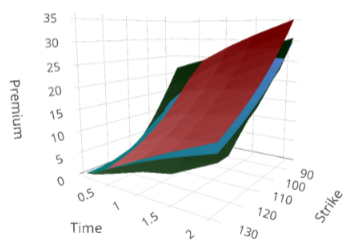


Figure 14: Microsoft Call Volatility Surface

MSFT Put Volatility Surface

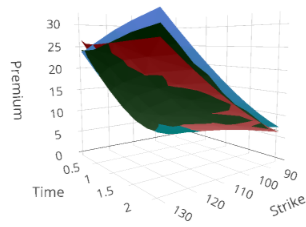


Figure 15: Microsoft Put Volatility Surface

SPY Call Volatility Surface

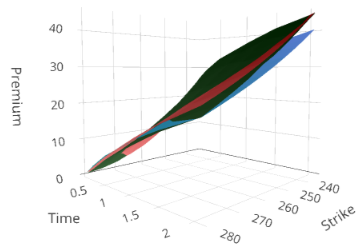


Figure 16: SPY Call Volatility Surface

SPY Put Volatility Surface

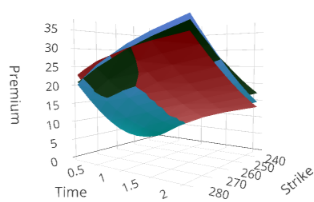


Figure 17: SPY Put Volatility Surface

V Call Volatility Surface

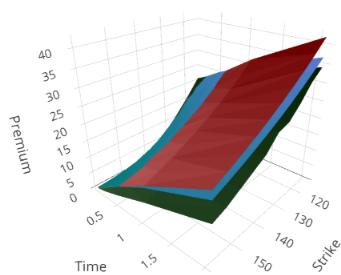


Figure 18: Visa Call Volatility Surface

V Put Volatility Surface

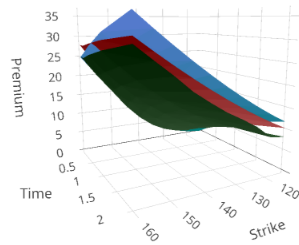


Figure 19: Visa Put Volatility Surface