QF620 Stochastic Modelling in Finance

Project Report

Chang Li-Yuan

Li Mengmeng

Lee Yu-Ching

Meng Qingshu

Shang Yuyang

Yang Xiaowei

Part I - Analytical Option Formulae

1. Black-Scholes Model

• Vanilla call/put

$$C = S\Phi(d_1) - Ke^{-rT}\phi(d_2)$$
$$P = -S\Phi(-d_1) + Ke^{-rT}\phi(-d_2)$$
$$d_1 = \frac{\log\frac{S}{K} + \left(r + \frac{\sigma^2}{2}\right)T}{\sigma\sqrt{T}}$$
$$d_2 = d_1 - \sigma\sqrt{T}$$

• Digital cash-or-nothing call/put

$$C_{Cash\ \ Digital} = e^{-rT}\Phi(d_2)$$
$$P_{Cash\ \ Digital} = e^{-rT}\Phi(-d_2)$$

• Digital asset-or-nothing call/put

$$C_{Asset\ \ Digital} = S\Phi(d_1)$$
$$P_{Asset\ \ Digital} = S\Phi(-d_1)$$

2. Bachelier Model

• Vanilla call/put

$$C = (S-K)\Phi(d_3) + S\sigma\sqrt{T}\phi(d_3)$$
$$P = (K-S)\Phi(-d_3) + S\sigma\sqrt{T}\phi(-d_3)$$
$$d_3 = \frac{S-K}{S\sigma\sqrt{T}}$$

• Digital cash-or-nothing call/put

$$C_{Cash\ \ Digital} = \Phi(d_3)$$
$$P_{Cash\ \ Digital} = \Phi(-d_3)$$

• Digital asset-or-nothing call/put

$$C_{Asset\ \ Digital} = S\Phi(d_3) + S\sigma\sqrt{T}\phi(d_3)$$
$$P_{Asset\ \ Digital} = S\Phi(-d_3) - S\sigma\sqrt{T}\phi(-d_3)$$

3. Black76 Model

• Vanilla call/put

$$C = e^{-rT}[F\Phi(d_1) - K\Phi(d_2)]$$
$$P = e^{-rT}[-F\Phi(-d_1) + K\Phi(-d_2)]$$
$$d_1 = \frac{\log\frac{F}{K} + \left(r + \frac{\sigma^2}{2}\right)T}{\sigma\sqrt{T}}$$

$$d_2 = d_1 - \sigma\sqrt{T}$$

• Digital cash-or-nothing call/put

$$C_{Cash\ \ Digital} = e^{-rT}\Phi(d_2)$$
$$P_{Cash\ \ Digital} = e^{-rT}\Phi(-d_2)$$

• Digital asset-or-nothing call/put

$$C_{Asset\ \ Digital} = e^{-rT}F\Phi(d_1)$$
$$P_{Asset\ \ Digital} = e^{-rT}F\Phi(-d_1)$$

4. Displaced-Diffusion Model

• Vanilla call/put

$$C = e^{-rT}[\frac{F}{\beta}\Phi(d_1) - (K + \frac{1-\beta}{\beta}F)\Phi(d_2)]$$

$$d_1 = \frac{\log\frac{F}{\beta K+(1-\beta)F} + \left(r + \frac{\sigma^2\beta^2}{2}\right)T}{\sigma\beta\sqrt{T}}$$

$$d_2 = d_1 - \sigma\beta\sqrt{T}$$

$$P = e^{-rT}[\frac{-F}{\beta}\Phi(-d_1) - (K + \frac{1-\beta}{\beta}F)\Phi(-d_2)]$$

• Digital cash-or-nothing call/put

$$C_{Cash\ \ Digital} = e^{-rT}\Phi(d_2)$$
$$P_{Cash\ \ Digital} = e^{-rT}\Phi(-d_2)$$

• Digital asset-or-nothing call/put

$$C_{Asset\ \ Digital} = e^{-rT}\frac{F}{\beta}\Phi(d_1)$$
$$P_{Asset\ \ Digital} = e^{-rT}\frac{F}{\beta}\Phi(-d_1)$$

Part II - Model Calibration

For each strike, the mid price is calculated as

$$\text{Mid Price} = \frac{\text{Best Bid} + \text{Best Ask}}{2}$$

Next, we calculate the forward price

$$F = S_0 e^{rT}$$

From the data provided, we know that the closing price for Google stock on 30-Aug-2013 is $846.9. Based on the r and T value we have calculated, we can calculate the forward value F is $851.7.

There is very limited liquidity on the in-the-money options -- market tends to trade at-the-money or out-of-the-money options. This means that we should focus on:

- High strikes calls（K ≥ F）
- Low strikes puts（K ≤ F）

Displaced-Diffusion Model Calibration

In practice, use the mid-price to find ATM at where the strike price is closest to the forward price (K=850). The ATM volatility is

$$\sigma = 0.2583$$

Using the "least_squares" function to calibrate the model, we can calculate the calibrated β (DDbeta) is

$$\beta = 0.3297$$

With the new parameter DDbeta, the calibrated Displaced-Diffusion Model implied volatility (impliedvol_DD) is generated.

Implied Volatility of the Calibrated Displaced-Diffusion Model

	strike	impliedvol_market	impliedvol_normal	impliedvol_lognormal	impliedvol_DD1	impliedvol_DD2	impliedvol_DD3	impliedvol_DD4	impliedvol_DD
0	320.0	0.393102	0.410059	0.258274	0.382252	0.354314	0.324730	0.293003	0.364307
1	340.0	0.361520	0.399618	0.258274	0.373432	0.347321	0.319742	0.290289	0.356655
2	350.0	0.365782	0.394686	0.258274	0.369270	0.344029	0.317402	0.289021	0.353048
3	360.0	0.355104	0.389931	0.258274	0.365260	0.340861	0.315155	0.287808	0.349577
4	370.0	0.348660	0.385342	0.258274	0.361392	0.337810	0.312995	0.286645	0.346232

SABR Model Calibration

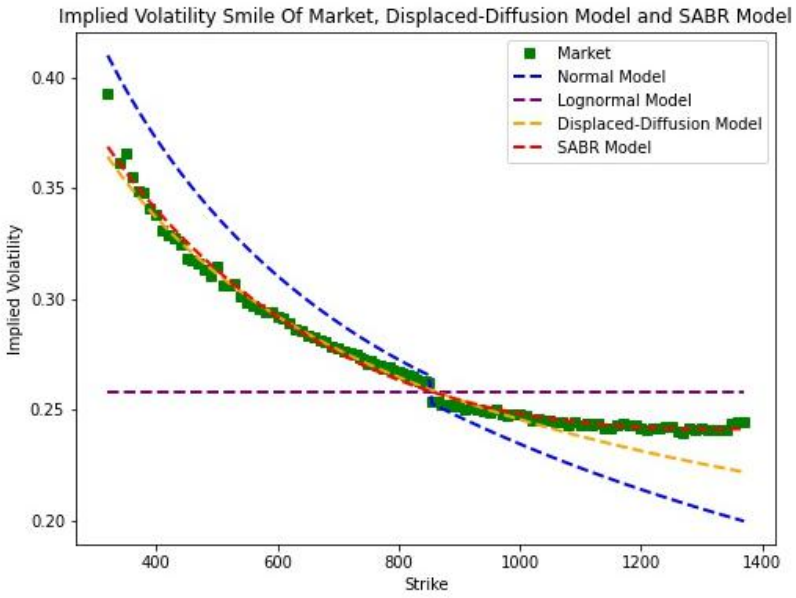
In order to match the output of the SABR model with the implied volatility observed in the market. We use the least_squares algorithm in "scipy" package to minimize the sum of squared error terms, i.e.

$$\min_{\alpha,\,\rho,\,\nu} \sum_{i=1}^{n} \epsilon_i^2$$

and calibrate the SABR model parameters:

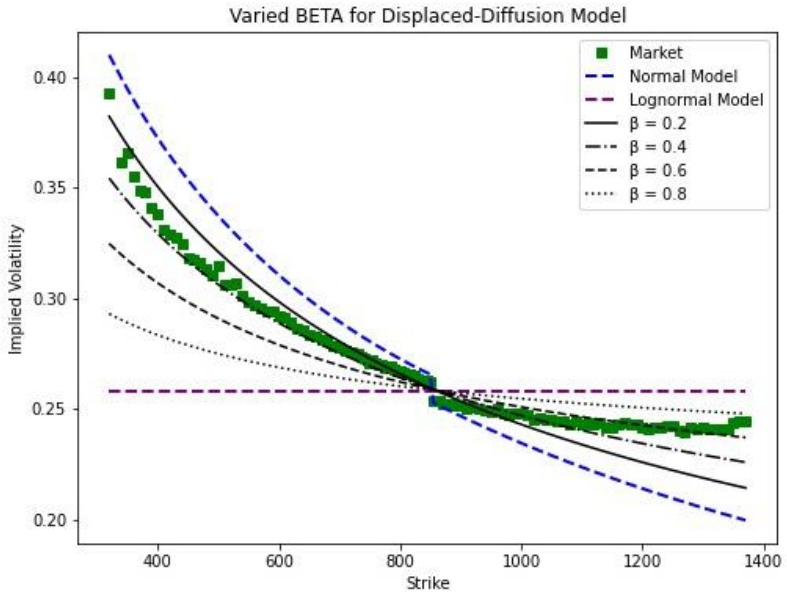
$$\alpha = 0.9908, \rho = -0.2851, \nu = 0.3522$$

The Fitted Implied Volatility Smile Against the Market Data



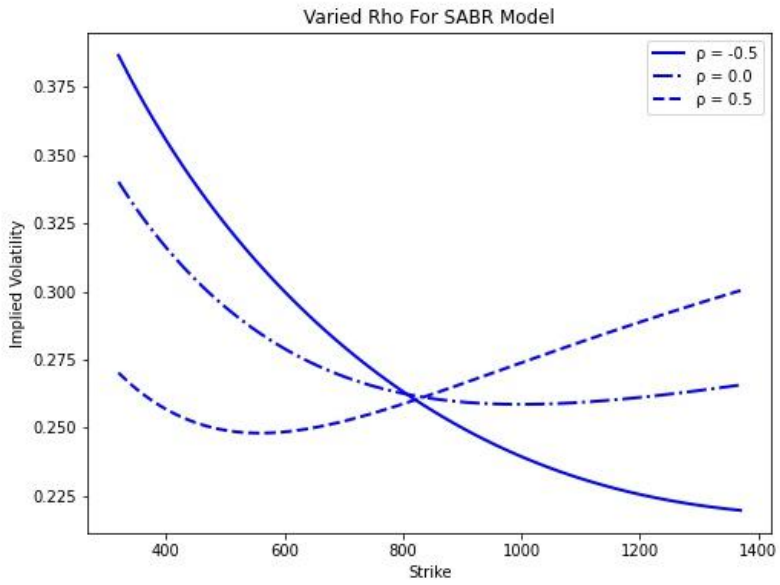
From the figure we can see that the SABR model fits the market model better than the displaced-diffusion model. This is because displaced-diffusion model can only fit to implied volatility skew - there will be mismatch if the implied volatility surface also exhibit "smile" characteristic. SABR model is able to fit both skew and smile in the implied volatility surface.

The Implied Volatility Smile of Displaced-Diffusion Model with Varied β



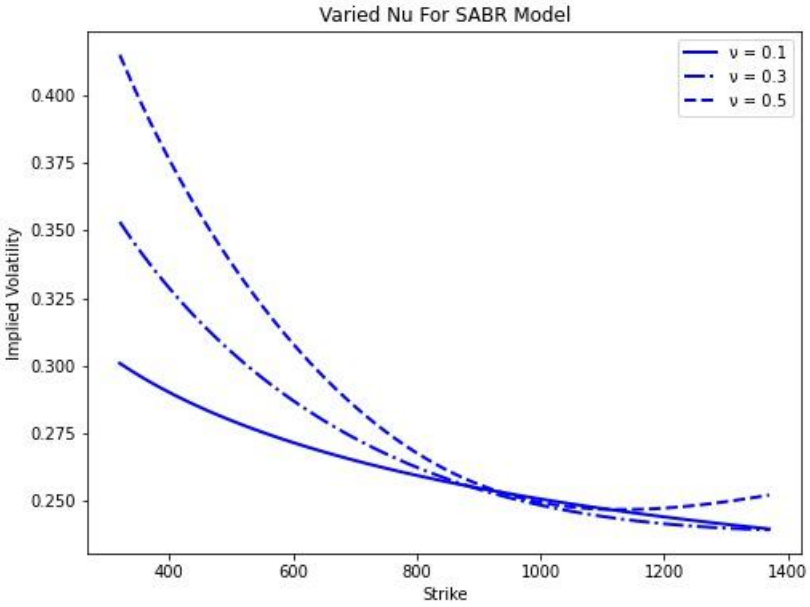
The range of beta is between 0 and 1. The smaller the beta, the closer the implied smile is to the normal model. If beta=0, it will be exactly equal to the Bachelier model. The higher the beta, the closer the implied smile is to the Lognormal model. If beta=1, it will be exactly equal to the Black model.

The Implied Volatility Smile of SABR Model with Varied ρ



The correlation parameter ρ is proportional to the skewness of stock returns. Negative correlation ($\rho<0$) increases the price of out-of-the-money put options and decreases the price of out-of-the-money call options. Positive correlation ($\rho>0$) decreases the price of out-of-the-money put options and increases the price of out-of-the-money call options.

The Implied Volatility Smile of SABR Model with Varied υ



The volatility-of-volatility υ has an impact on the kurtosis and proportional skewness of stock returns. Generally, larger volatility-of-volatility υ increases the price of out-of-the-money call and put options.

Part III - Static Replication

On part3, our objective is to determine the price under two pricing model of European derivatives which have payoff function as following:

$$S_T^3 \times 10^{-8} + 0.5 \times \log(S_T) + 10$$

To work out our problem using python, we divide our part into 3 steps:

1. Derive price formula under Black-Scholes Model and Bachelier Model

(1) Under Black-Scholes model

Derive process:

According to payoff function, there is no condition for contract holder to obtain payoff, because under Black-Scholes Model, stock price would never be negative.

$$S_T = S_0 e^{\left(r-\frac{\sigma^2}{2}\right)T+\sigma\sqrt{T}x}, x \sim N\ (0,1) \quad \text{......①}$$

$$log(s_T) = log(s_0) + \left(r - \frac{\sigma_2}{2}\right) T + \sigma\sqrt{T}x \quad \text{......②}$$

V_t denote the value of this financial contract at time *t,* under the Q^*measure, we have: $\frac{V_0}{B_0} = E^Q\,[\frac{V_t}{Bt}]$

$$V_0 = e^{-r_T} E^Q[S_T^3 \times 10^{-8} + 0.5 \times log(S_T) + 10] \quad \text{......③}$$

Bring ①② into ③, we obtain final result:

$$V_0 = e^{-rT}[S_0^3 \times e^{3(r-\frac{\delta^2}{2})} \times 10^{-8} \times e^{\frac{9}{2}\delta^2 T} + \frac{1}{2}\log(S_0) + \frac{1}{2}(r - \frac{\delta^2}{2})T + 10]$$

(2)Under Bachelier model

Derive process:

$$S_T = S_0 + \sigma S_0 w_T = S_0\left(1 + \sigma\sqrt{T}x\right), x \sim N\ (0,1) \quad \text{......④}$$

$$log(S_T) = log(S_0) + log\left(1 + \sigma\sqrt{T}x\right) \quad \text{......⑤}$$

Under Bachelier model, the stock price could become negative. Therefore, according to payoff function, the probability of earning payoff should be as follow:

$$P\{s_T > 0\} = P\{s_0(1+\sigma\sqrt{T}x) > 0\} = P\{(1+\sigma\sqrt{T}x) > 0\} = P\left\{x > \frac{-1}{\sigma\sqrt{T}}\right\}$$

Let $x^* = \frac{-1}{\sigma\sqrt{T}}$

V_t denote the value of this financial contract at time *t,* under the Q^*measure, we have: $\frac{V_0}{B_0} = E^Q\,[\frac{V_t}{Bt}]$

$$V_0 = e^{-r_T} E^Q[S_T^3 \times 10^{-8} + 0.5 \times log(S_T) + 10] \quad \text{......③}$$

Bring ④⑤ into ③, we obtain:

$$V_0 = e^{-r_T} E^Q[0.5 \times log(s_0) + 10] + e^{-rT} E^Q\left[S_0^3\left(1+\sigma\sqrt{T}x\right)^3 \times 10^{-8}\right] + e^{-rT} E^Q\left[\frac{1}{2} \times log\left(1+\sigma\sqrt{T}x\right)\right]$$

Where:

$$f_1(x) = e^{-rT}E^Q\left[S_0^3\left(1+\sigma\sqrt{T}x\right)^3 \times 10^{-8}\right] = S_0^3 \times 10^{-8}\int_{x^*}^{+\infty}\left(1+\sigma\sqrt{T}x\right)^3 \frac{1}{\sqrt{2\Pi}}e^{-\frac{x^2}{2}}dx$$

$$f_2(x) = E^Q\left[\frac{1}{2}log\left(1+\sigma\sqrt{T}x\right)\right] = \frac{1}{2}\int_{x^*}^{+\infty} log\left(1+\sigma\sqrt{T}x\right)\frac{1}{\sqrt{2\Pi}}e^{-\frac{x^2}{2}}dx$$

Final result:

$$V_0 = e^{-rT}[\frac{1}{2}\log(S_0)+10] + e^{-rT} \times S_0^3 \times 10^{-8} \times f_1(x) + e^{-rT} \times \frac{1}{2} \times f_2(x)$$

2. After deriving the calculation formula, we are supposed to determine values of lasting period(T), discount rate(r), start value(S0) and the parameter(δ).

(1) T=1.38

The contract starts from 30-Aug-2013 and expire on 17-Jan-2015, so the lasting period can be calculated using python, turning out to be 505 days or 1.38 year.

(2) r= 0.0042

The file "discount.csv" contains information about the "zero rates" to be used for discounting. For instance, to discount a cash flow paid 3400 days from today, the discount factor is $D(0,T) = e^{-0.0313462 \times \frac{3400}{365}}$.

Although our payment date is not provided we can interpolate for the zero rate. In the other word, we use linear interpolation to solve our discount rate. T=505 days is in between 474 days and 565 days,

so r=(505-474)/(565-474)*r1+(565-505)/(565-474)*r2=0.42%

index	Day	Rate (%)
11	474	0.38802
12	565	0.43892

(3) S0 is given as $846.9.

(4) δ=0.258

Since we only consider at the money situation because it is the most liquid, sigmas(δ) are the same under Black-Scholes Model and Bachelier Model as well as SABR Model. So we can directly use sigma(around 0.258) plotted in part2 here.

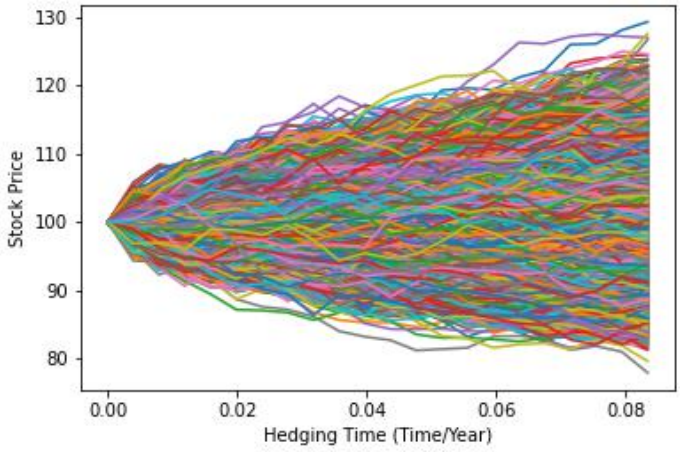
3. Now it is time to put the calculated value into our defined formulas.

Results are presented here:

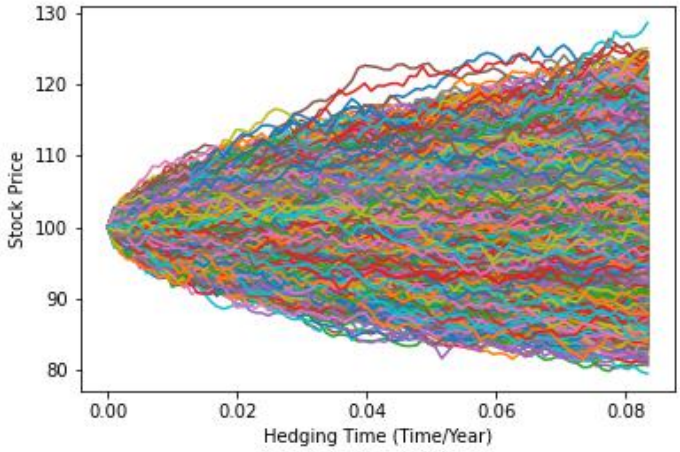
	Black-Scholes	Bachelier
price	21.368467	20.969861

Part IV-Dynamic Hedging

In this part, we use Black-Scholes model to simulate the stock price. Suppose we sell this at-the-money call option, and we hedge N times during the life of the call option. We use 50,000 path in our simulation and plot the histogram of the hedging error for N=21 and N=84.



(50,000 path, N=21)

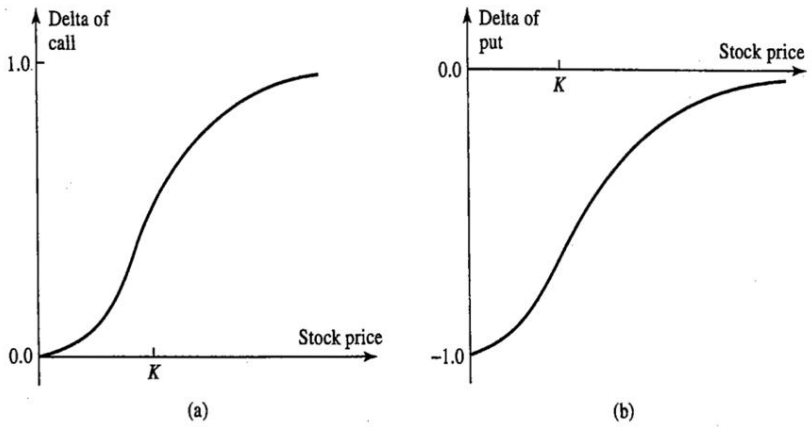


(50,000 path, N=84)

The Black-Scholes Formula equation does assume that there is no profit to be made with its replicating strategy. However, it isn't designed to give you a profitable trading strategy but is meant to give a no arbitrage price for the option. Therefore, since we don't know what the realized volatility will be. The performance of your hedge depends not only on whether the realized volatility is higher or lower than your estimate but also on whether the market is range bound or trending.

Therefore, since the market will either range up and down, the delta is changing all the time. Delta measures the sensitivity of the option to the stock price.

Figure Variation of delta with stock price for (a) a call option and (b) a put option on a non-dividend-paying stock.

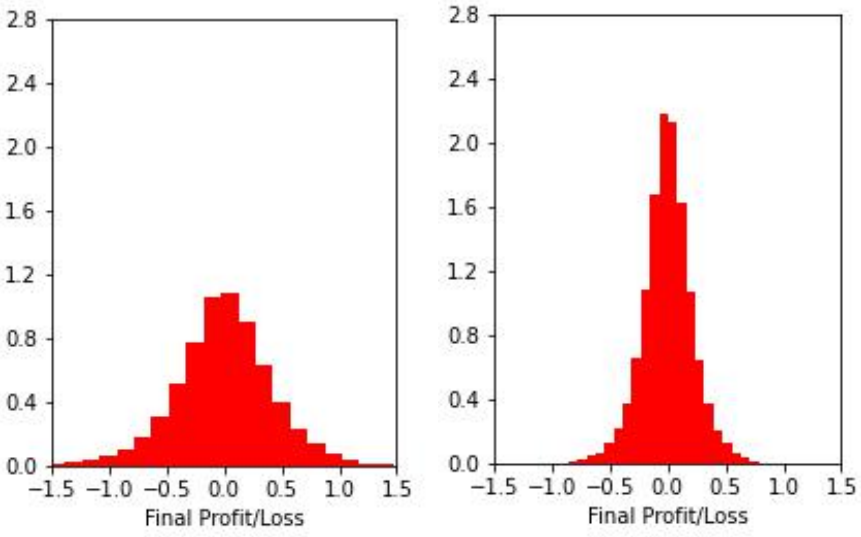


Therefore, since the delta is changing all the time, the offset Δ -Hedging can only be maintained for a very short period of time, which means that Δ -Hedging needs to be continuously adjusted, which is called Rebalancing.

Generally speaking, hedging strategies that are constantly adjusted over time are collectively referred to as Dynamic Hedging, and what is shown here is the use of Δ for hedging.

However, only if we split the time to a very low period, the Δ - Hedging can be perfect hedging otherwise is not perfect hedging and will suffer profit and loss.

The below image is the hedging error for N=21 and N=84. We can conclude that if we hedge more times during the life of a call option, the final P&L of our dynamic hedging strategy will concentrate to zero, which is exactly the same results as we previous discussion of Dynamic Hedging.



N=21 N=84