M.Sc. in Financial Engineering
Master Level M2 – Academic Year 2025–26
Professor: Dr. Jeroen Kerkhof

ASSIGNMENT (Python Project)
ADVANCED DERIVATIVES

Prepared by:

KARADAS Baris
Nicholas Marsh
CHEVRIER Clément

Contents

1 Black-Scholes-Merton Hedging

1.1 FinancePy

Install `FinancePy` and for speed issues, in the following, call directly into the model library rather than go via the `EquityOptions` class.

1.2 DeltaHedge

Write a Python function called `DeltaHedge`. The dynamics of the stock price should be assumed to be lognormal with a drift μ and a volatility σ_{Real}. The output of your function should be a tuple that has 4 elements: The terminal stock price $S(T)$ in the simulation path; The option payoff; The realized variance of returns during the hedging period; The replicating error which is the difference between the total value of the hedging portfolio and the option payoff.

1.3 DeltaHedge (Extension)

Write another function that calls the previous `DeltaHedge` function and which can then be used to calculate the hedging error over 10,000 different paths.

1.4 Hedging Error

Consider a put option with $S(0) = 100$, $K = 100$, $r = 4\%$, $T = 1.0$ and $\sigma = 20\%$. Assume here that $\mu = 5\%$. For this option, make a scatterplot of the hedging error (y-axis) versus the terminal stock price (x-axis) for $N = 12$ (monthly), $N = 52$ (weekly) and $N = 252$ (daily). Use different symbols or colours to distinguish the points.

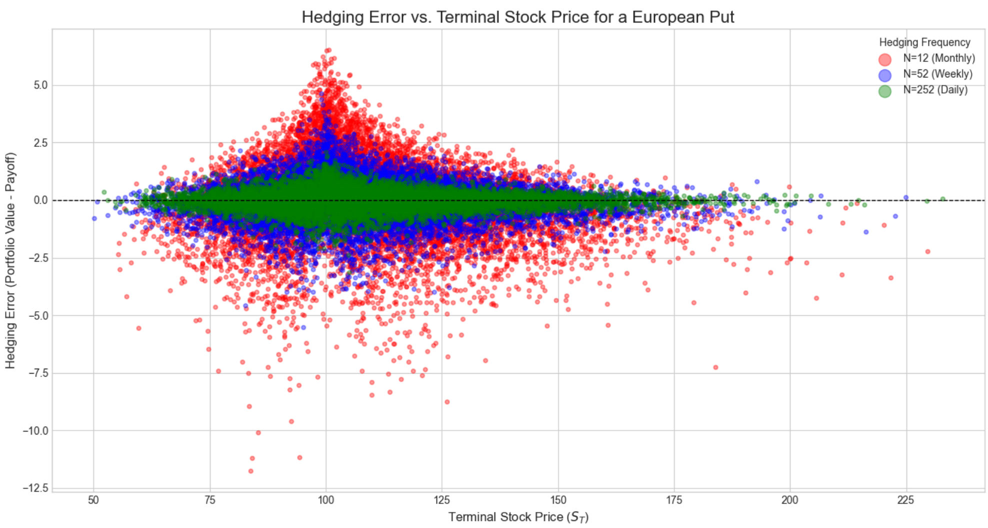

Figure 1: Scatterplot of the hedging error versus terminal stock price for different hedging frequencies.

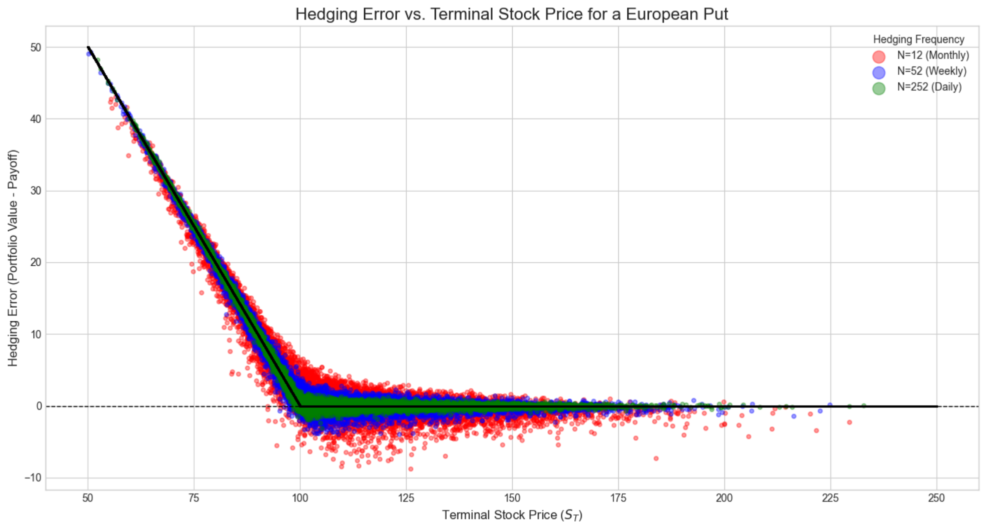

Figure 2: Hedging Error as a Function of Terminal Stock Price for a European Put.

1.5 Variance of Hedging Error

For each value of N also calculate the mean and variance of this option hedging error over 10,000 different paths. You can use this to generate the answers to the remaining parts of this question. Present this in a simple table format.

Table 1: Mean and Variance of the Hedging Error for Different Hedging Frequencies

Hedging Frequency	**Mean Hedging Error**	**Variance of Hedging Error**
$N = 12$ (Monthly)	-0.0164	3.5067
$N = 52$ (Weekly)	-0.0197	0.8587
$N = 252$ (Daily)	-0.0048	0.1837

Comment: The results show that as the hedging frequency increases, both the mean and variance of the hedging error decrease substantially. When rebalancing daily ($N = 252$), the variance of the hedging error is more than an order of magnitude smaller than in the monthly case, indicating a more accurate replication of the option's payoff. This behavior is consistent with the theoretical expectation that a higher rebalancing frequency reduces discretization error in the delta-hedging strategy. The small negative mean error across all frequencies suggests a slight tendency to under-hedge, possibly due to transaction timing or discretization effects in the simulated paths. To get rid of the error we should hedge at a continuous time.

1.6 Realized Volatilities vs Replication error

Create a scatterplot of the realized volatilities vs the replication error. Explain the pattern.

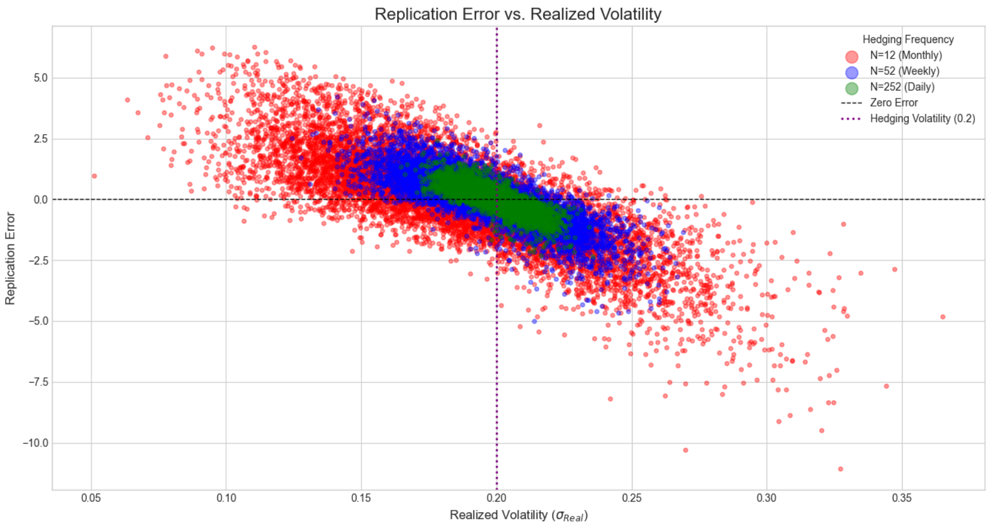

Figure 3: Replication Error vs Realized Volatility.

The replication error of a delta-hedged option arises from two sources:

$$\text{Error} \approx \frac{1}{2}\int_0^T \Gamma_t S_t^2 \left(\sigma_{\text{real}}^2 - \sigma^2\right) dt$$

where $\Gamma_t = \frac{\partial^2 V^{\text{BS}}}{\partial S^2}(S_t, t)$. Hence, the sign and magnitude of the error are directly driven by the volatility mismatch $(\sigma_{\text{real}}^2 - \sigma^2)$ and the option's convexity.

Key interpretation.

- If $\sigma_{\text{real}} > \sigma$, then $\sigma_{\text{real}}^2 - \sigma^2 > 0$ and the error is **positive**: realized volatility exceeded the hedged volatility, so the hedge was too short gamma (under-hedged).
- If $\sigma_{\text{real}} < \sigma$, the term is negative and the hedge is **over-hedged**, producing a negative replication error.

The scatterplot of replication error vs. realized volatility is roughly **linear**: positive slope crossing zero at $\sigma_{\text{real}} = \sigma = 20\%$. This reflects the proportional relationship in the integral above. As the rebalancing frequency N increases:

$$N = 12 \;\Rightarrow\; \text{large variance}, \qquad N = 252 \;\Rightarrow\; \text{small variance}.$$

Higher N reduces discretization error $\mathcal{O}(\sqrt{\Delta t})$, but cannot remove model error from a wrong σ. Error sign: $\text{sgn}(\text{Error}) = \text{sgn}(\sigma_{\text{real}}^2 - \sigma^2)$. Dispersion $\downarrow$ as $N \uparrow$. Continuous-time limit $\Rightarrow$ error $= 0$ if $\sigma_{\text{real}} = \sigma$. Thus, the plot slopes upward, narrowing as N increases: hedging frequency controls *variance*, volatility mis-specification controls *bias*.

1.7 Impact of the Drift μ on Hedging Error

For the same put option, calculate the mean absolute error value and the variance of the hedging error for $\mu = 2.5\%, 5.0\%, 7.5\%, 10\%$ by sampling 10,000 hedging paths using $N = 52$. Show the results in a table. What does this tell you? Does the value of the drift change the hedging by a little or a lot?

We evaluate how the real-world drift μ affects the hedging error of a European put under weekly rebalancing ($N = 52$). The simulated drifts are

$$\mu \in \{2.5\%, 5.0\%, 7.5\%, 10.0\%\}.$$

The mean absolute error (MAE) and variance of the hedging error were computed over 10,000 paths:

Table 2: Effect of Drift μ on the Hedging Error ($N = 52$)

Real-World Drift μ	**Mean Error**	**Variance of Hedging Error**
2.5%	-0.00065	0.9352
5.0%	-0.00632	0.9016
7.5%	-0.00469	0.9232
10.0%	0.00024	0.8520

Mathematical insight. In the BSM PDE,

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV = 0,$$

the drift μ of the real-world process does *not* appear. Delta hedging eliminates first-order exposure to the stock's direction, thus, the replication error mainly depends on volatility, not drift. Any effect from μ appears only through the discrete sampling of stock paths.

Interpretation. The table shows that increasing μ has almost no effect on the mean absolute error or its variance. This is because:

- Hedging neutralizes μ by dynamically adjusting $\Delta_t = \frac{\partial V^{\text{BS}}}{\partial S}$;
- The small residual differences arise from path-dependent discretization errors.

Conclusion.

- μ has a **minor impact** on hedging performance.
- Hedging error is dominated by **volatility mis-specification** and **rebalancing frequency**.

$$\boxed{\text{In practice: } \Delta\text{Hedging Error from } \mu \ll \Delta\text{Hedging Error from } \sigma}$$

2 Time-Varying Volatility

2.1 Adjustment to Time-Varying Volatility

We start with the code from Question 1. Instead of using a constant volatility of 20%, the volatility is now time-dependent: 4% during the first half of T, and 28% during the second half. Value a European put option with:

$$S(0) = 100, \quad K = 100, \quad r = 4\%, \quad T = 1.0.$$

Compare this to the value of the same put option under a constant volatility $\sigma = 20\%$.

Over $[0, T]$ the option price depends on the *integrated variance.* For this profile the total variance is:

$$\int_0^T \sigma(t)^2\,dt = \int_0^{1/2} 0.04^2\,dt + \int_{1/2}^1 0.28^2\,dt = \tfrac{1}{2}(0.04^2 + 0.28^2) = \tfrac{1}{2}(0.0016 + 0.0784) = 0.0400.$$

Hence the *effective* (average) volatility is:

$$\sigma_{\text{eff}} = \sqrt{0.0400} = 0.20.$$

Since the Black–Scholes price for a European option with deterministic volatility depends on the integrated variance only, both specifications, constant $\sigma = 20\%$ and time-varying $(4\%, 28\%)$ — imply the same variance 0.04 over $[0, 1]$, and therefore

$$P_{\text{const}} = P_{\text{TV}}.$$

This is exactly what the code output shows: the put price with constant volatility and the put price with the two-step volatility are numerically identical (up to machine precision). The hint in the assignment ("there should be no difference") is therefore a direct consequence of matching the total variance.

2.2 Updated Delta Hedging Function

Create a modified delta hedging function, `DeltaHedgeTV`, that accounts for **time-varying volatility**. The function should dynamically adjust the volatility input $\sigma(t)$ according to the time elapsed:

$$\sigma(t) = \begin{cases} 0.04 & \text{for } 0 \le t < \frac{T}{2}, \\ 0.28 & \text{for } \frac{T}{2} \le t \le T. \end{cases}$$

2.3 Delta Hedge Performance Comparison

Repeat the delta-hedging exercise from Question 1(d) using the time-varying volatility function. Generate a scatterplot of the hedging error (y-axis) versus terminal stock price (x-axis) for $N = 12, 52, 252$. How does the performance of the hedge compare with the constant volatility case?

Setup. We keep the market (true) volatility piecewise–constant,

$$\sigma_{\text{real}}(t) = \begin{cases} \sigma_1 = 0.04, & 0 \le t < T/2, \\ \sigma_2 = 0.28, & T/2 \le t \le T, \end{cases}$$

and, *as in the code*, we hedge with the **forward** (remaining) volatility obtained from the deterministic term structure:

$$\sigma_{\text{fwd}}(t) = \sqrt{\frac{1}{T-t}\int_t^T \sigma^2_{\text{hedge}}(u)\,du} = \begin{cases} \sqrt{\dfrac{\sigma_1^2\,(T/2-t) + \sigma_2^2\,(T/2)}{T-t}}, & t < T/2, \\ \sigma_2, & t \ge T/2, \end{cases}$$

so that $\sigma_{\text{fwd}}(0) = \sqrt{(\sigma_1^2 T/2 + \sigma_2^2 T/2)/T} = 0.20$, then $\sigma_{\text{fwd}}(t) \uparrow 0.28$ as $t \ge T/2$, and it stays at 0.28 thereafter. The hedge delta at time t is $\Delta^{\text{TV}}(S_t, t)$ computed with $\sigma_{\text{fwd}}(t)$. We simulate 10,000 paths; $N \in \{12, 52, 252\}$ re-hedges.

Continuous–time benchmark. With deterministic $\sigma(t)$ and a model that *matches* it, the BS PDE holds and *continuous* delta hedging replicates exactly. This removes the model–mismatch (bias) term; only **discretization** error remains.

Numerical results (yours) and comparison to Q1.4.

	Time–varying (forward hedge)		**Q1.4 constant** 20%	
Hedging Freq.	Mean	Variance	Mean	Variance
N=12 (Monthly)	−0.027366	6.821015	−0.016438	3.506675
N=52 (Weekly)	0.010765	1.738329	−0.019651	0.858727
N=252 (Daily)	−0.003216	0.365270	−0.004847	0.183730
Variance ratio	1.95×		(N=12)	
	2.03×		(N=52)	
	1.99×		(N=252)	

What we see:

- Means are tiny relative to their standard deviations (|mean|/stdev $\ll 1$): the hedge is essentially *unbiased*, as expected when using the correct forward vol.
- Variances are about **twice** those of Q1.4 for each N. The ordering with N is unchanged: daily < weekly < monthly.

Why the variance is higher even with a correct (forward) vol. With the bias removed, the leading discrete–time error for delta hedging scales like

$$\text{Var(Error)} \;\approx\; C\,\Delta t\,\mathbb{E}\left[\int_0^T \Big(\Gamma_t S_t \sigma(t)\Big)^2\,dt\right],$$

for a model–dependent constant C. Hence, the *time profile* of $\sigma(t)$ matters through the weight Γ_t^2. In our experiment the high volatility 0.28 is concentrated in the *second* *half*, precisely when the option's gamma is largest. This increases $\int (\Gamma S\sigma)^2 dt$ versus the flat–20% case with the same total integrated variance $\int_0^T \sigma^2(t)\, dt = 0.04$. Result: **larger** discretization risk (roughly 2× in your runs), even though the hedge uses the correct forward vol and the mean error is near zero.

Shape of the scatter (vs. Q1.4). The error vs. S_T cloud keeps the triangular/funnel shape. Relative to Q1.4 it is noticeably thicker near-the-money and in the upper wing, consistent with late high vol and high gamma amplifying discrete re-hedging slippage. The ranking N=252 < N=52 < N=12 is preserved.

Trading interpretation. When the volatility term structure is known/deterministic and you delta–hedge with the *forward* (remaining) vol, you eliminate model bias but you still face **timing risk of variance**: concentrating variance during the high–gamma window magnifies discrete hedging P&L. Increasing N mitigates but cannot eliminate this effect.

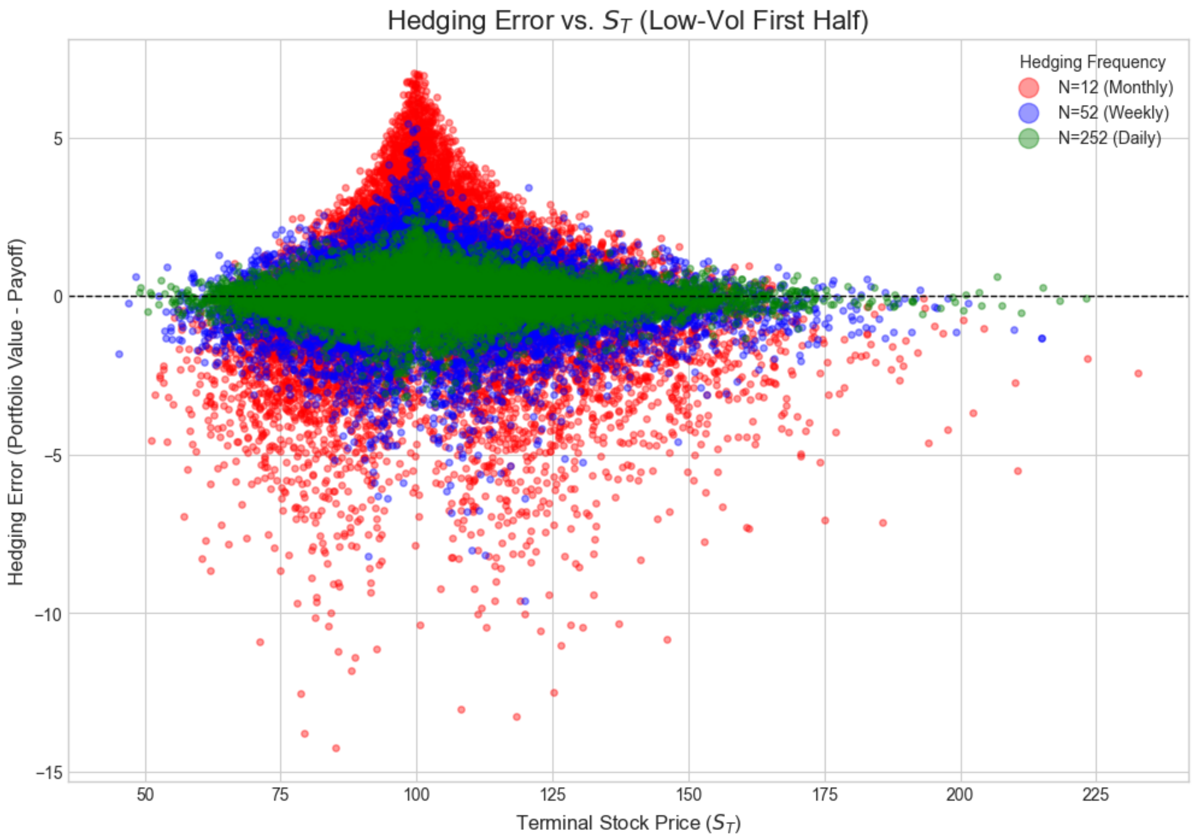

Figure 4: Hedging error vs. terminal stock price for a time-varying realized volatility path (4%→28%), with the hedge priced and rebalanced using the *forward* (remaining).

2.4 Reversed Volatility Regime

Now reverse the volatility structure: set the volatility to 28% in the first half of the option's life and 4% in the second half:

$$\sigma(t) = \begin{cases} 0.28 & \text{for } 0 \leq t < \frac{T}{2}, \\ 0.04 & \text{for } \frac{T}{2} \leq t \leq T. \end{cases}$$

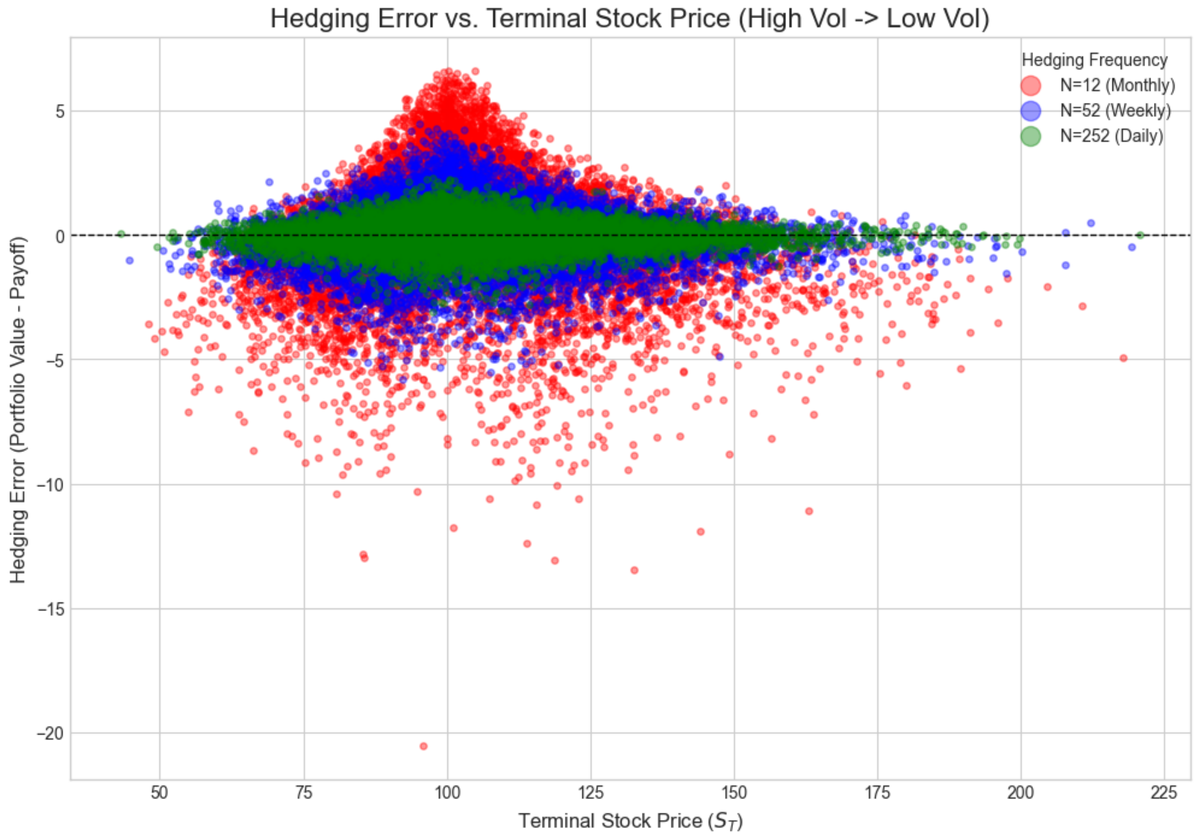

Figure 5: Hedging error vs. terminal stock price for a time-varying realized volatility path (28%→4%), with the hedge priced and rebalanced using the *forward* (remaining).

2.5 Discussion and Interpretation

Setting. In the two time–varying regimes ("Low→High" and "High→Low"), the *market* follows $\sigma_{\text{real}}(t) \in \{4\%, 28\%\}$, while the hedge uses the *forward* Black–Scholes volatility

$$\sigma_{\text{fwd}}(t) = \sqrt{\frac{1}{T-t}\int_t^T \sigma_{\text{real}}^2(u)\,du},$$

so pricing and deltas at each rebalance reflect the *remaining* integrated variance. We contrast these results to Q1.4 (constant 20% market & hedge).

Table 3: Hedging error statistics (10,000 paths). Forward–vol hedge for time–varying regimes; Q1.4 is constant 20%.

Scenario	Frequency	Mean Error	Variance of Error
1. Constant Vol (20%)	N=12 (Monthly)	−0.0164	3.5067
	N=52 (Weekly)	−0.0197	0.8587
	N=252 (Daily)	−0.0048	0.1837
2. Low→High (4% → 28%)	N=12 (Monthly)	−0.0274	6.8210
	N=52 (Weekly)	0.0108	1.7383
	N=252 (Daily)	−0.0032	0.3653
3. High→Low (28% → 4%)	N=12 (Monthly)	0.0329	6.5989
	N=52 (Weekly)	0.0093	1.5913
	N=252 (Daily)	−0.0060	0.3295

What changes when we hedge with forward vol? With deterministic $\sigma(t)$ and forward-vol hedging, the *model-mismatch* (bias) term vanishes, and the residual error is due to discrete rebalancing:

$$\underbrace{\frac{1}{2}\int_0^T \Gamma_t S_t^2\Big(\sigma_{\text{real}}^2(t)-\sigma_{\text{fwd}}^2(t)\Big)dt}_{=0} \quad + \quad \underbrace{\text{discretization noise}}_{\approx\; C\,\Delta t\,\mathbb{E}\int_0^T \left(\Gamma_t S_t\,\sigma_{\text{real}}(t)\right)^2 dt}.$$

Consistent with this, the sample *means* in Table 3 are tiny relative to their standard deviations: the hedge is essentially unbiased.

How do the variances compare to Q1.4? Even with the correct (forward) σ, the **variance** of the error is *larger* than in Q1.4 at the same N:

$$\mathrm{Var}^{\text{Low}\to\text{High}}/\mathrm{Var}^{\text{Const}} \approx \begin{cases} 1.95\times, & N{=}12,\\ 2.03\times, & N{=}52,\\ 1.99\times, & N{=}252, \end{cases} \qquad \mathrm{Var}^{\text{High}\to\text{Low}}/\mathrm{Var}^{\text{Const}} \approx \begin{cases} 1.88\times, & N{=}12,\\ 1.85\times, & N{=}52,\\ 1.79\times, & N{=}252. \end{cases}$$

Why? Discrete hedging error weights *where* variance is realized through $(\Gamma S\sigma)^2$. Concentrating volatility when Γ is large amplifies slippage.

Low→High vs. High→Low. Placing high volatility *late* (4%→28%) is consistently **worse** than placing it early (28%→4%): variances are higher at every N (6.8210>6.5989, 1.7383>1.5913, 0.3653>0.3295). Intuition: near maturity gamma peaks, so late high vol magnifies $\int(\Gamma S\sigma)^2 dt$ more than early high vol, even though $\int_0^T \sigma^2(t)\,dt$ is the same in the two regimes.

Effect of N. Increasing N still helps substantially (e.g., Low→High: 6.8210 → 0.3653 from monthly to daily). But unlike Q1.4, the variance does *not* fall to the constant-vol benchmark because the "when-variance-arrives" effect survives any finite rebalancing grid.

Shape of the scatter. The error vs. S_T cloud remains a symmetric "funnel", centered near zero. Relative to Q1.4 it is *thicker* around-the-money (where Γ is largest), with the Low→High regime the thickest of all—a visual imprint of late high volatility.

Practical takeaway. Forward-vol hedging removes bias but not timing risk:

$$\boxed{\text{Correct level of variance} \not\Rightarrow \text{small hedging error.}}$$

What matters is *when* variance shows up. If high vol coincides with the high-Γ window, discrete hedging P&L dispersion can be almost $2\times$ the constant-vol case even with perfect knowledge of the future term structure. Operationally: if you anticipate a back-loaded vol regime, either (i) rebalance more frequently, (ii) overlay gamma (e.g., short-dated options) across the expected shock window, or (iii) de-risk the book earlier; otherwise you are long discretization risk.

One-line comparison to Q1.4. Q1.4 (flat 20%) is the best case: only discretization, uniformly spread in time. With forward vol under time-varying regimes we are still unbiased, but dispersion is $\approx$ 1.8–2.0$\times$ larger because the variance is *time-clustered* where Γ is highest.

3 Determining the Implied Density from the Volatility Skew

3.1 Fitting the Volatility Smile and Extracting the Implied Density

Suppose that we have managed to fit the 1-year volatility smile of the equity option market using the functional form:

$$\sigma(x) = ax^2 + bx + c,$$

where x is the *moneyness*, defined as $x = \frac{K}{S(0)}$. The parameters are given as:

$$a = 0.025, \quad b = -0.23, \quad c = 0.55,$$

and the initial stock price is $S(0) = 100$.

The volatility smile reflects how the market prices options across strikes when the Black–Scholes assumption of constant volatility fails. Instead of a flat volatility surface, empirical option markets exhibit a downward-sloping skew: deep out-of-the-money (OTM) puts trade with higher implied volatility than calls, indicating that the market anticipates more probability mass in the left tail (i.e., large downward moves in S_T).

Theoretical foundation. Under risk-neutral valuation, the price of a European call is

$$C(K,T) = e^{-rT} \int_K^\infty (S_T - K) f(S_T)\, dS_T,$$

where $f(S_T)$ is the risk-neutral density of the terminal stock price. Differentiating twice with respect to strike K yields the **Breeden–Litzenberger** (1978) relationship:

$$f(K) = e^{rT} \frac{\partial^2 C(K,T)}{\partial K^2}.$$

Hence, by observing call prices across strikes — or equivalently their implied volatilities one can recover the entire risk-neutral probability distribution implied by the market.

The function fitted to the implied volatility smile is

$$\sigma(x) = ax^2 + bx + c, \qquad \text{where} \quad a = 0.025, \quad b = -0.23, \quad c = 0.55, \quad x = \frac{K}{S_0}.$$

This quadratic form captures the negative slope typical of equity volatility skews, where $\sigma(x)$ decreases with strike K.

Interpreting the graph. Figure 6 compares:

- the **volatility smile** (red dashed line) fitted from market data;
- the corresponding **implied risk-neutral density** $f(K)$ (blue curve) obtained via the Breeden–Litzenberger formula;
- a **Gaussian reference distribution** (green curve) assuming constant volatility equal to the at-the-money (ATM) value, i.e. the standard Black–Scholes case.

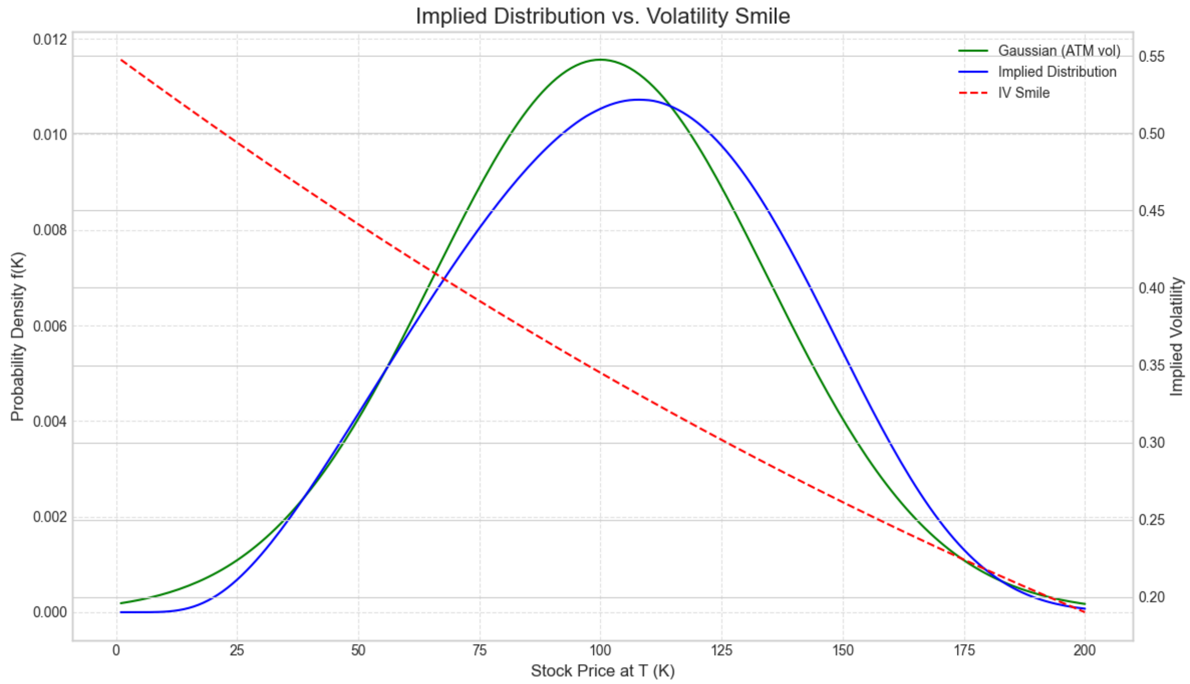

Figure 6: Implied distribution (blue) derived from the volatility smile (red dashed) compared with the Gaussian distribution implied by a flat volatility (green).

The red dashed line slopes downward, indicating that implied volatility is higher for low strikes and lower for high strikes. This corresponds to the empirical **equity skew**: market participants demand higher option premia (higher vol) for downside protection.

Consequently, the blue implied density deviates from the symmetric Gaussian benchmark:

- it is **left-skewed** (the left tail is thicker than the right);
- the **mode** of the distribution shifts slightly to the left of the initial price $S_0 = 100$;
- the **right tail** is thinner, implying smaller probabilities for large upward moves.

This left skew directly reflects risk aversion and the asymmetric payoff profile in equity markets: investors are willing to pay more for downside protection, which inflates implied volatilities for low strikes.

Interpretation. The comparison between the blue and green curves shows how a simple volatility smile transforms the shape of the implied density: a downward-sloping $\sigma(K)$ increases the left tail probability and compresses the right tail, producing a realistic, negatively skewed terminal distribution. The smile thus encodes the market's collective beliefs about fat tails and crash risk that the lognormal Black–Scholes world cannot capture.

3.2 Pricing a Digital Call Option Using the Implied Density

A digital call option pays \$1 if $S(T) > K$ and zero otherwise. Using the implied probability density $f(K)$ derived from the volatility smile in Section 3.1, compute the prices of 1-year European digital call options with strikes:

$$K = 60, 80, 100, 120, 140.$$

Assume the same market conditions:

$$r = 4\%, \qquad q = 0.$$

Construct a table of digital option prices across strikes.

Using the risk–neutral density extracted in 3.1 and discounting at $r = 4\%$, the 1-year European digital call option with strike K is priced as

$$\mathrm{DigCall}(K) = e^{-rT}\,\mathbb{Q}(S_T > K) = e^{-rT}\int_K^{\infty} f(s)\,ds,$$

where $f(\cdot)$ is the smile–implied density. Evaluating this for the strikes required in the assignment gives:

Table 4: Digital call prices implied by the volatility smile ($r = 4\%$, $T = 1$)

Strike K	**Digital Call Price**
60	0.8551
80	0.7150
100	0.5276
120	0.3238
140	0.1504

The prices decrease monotonically with K, as expected, but they do so *more slowly* on the downside (high price at $K = 60$) than a flat-vol Black–Scholes model would predict. This is the direct effect of the downward-sloping smile: higher implied vol for low strikes $\Rightarrow$ more left-tail probability $\Rightarrow$ higher digital prices for low K. At higher strikes ($K = 120, 140$) the smile assigns lower volatilities, so the digital prices fall faster.

3.3 Pricing Digital Call Options Using the Black–Scholes Formula

Using the same set of strikes and parameters, price the digital call options with the standard **Black–Scholes formula** for digital calls:

$$C_{\text{dig}}(K, T) = e^{-rT} N(d_2),$$

where

$$d_2 = \frac{\ln\left(\frac{S_0}{K}\right) + (r - q - \frac{1}{2}\sigma^2)T}{\sigma\sqrt{T}}.$$

Compare the results of this section to those obtained in Section 3.2.

Here we use the *implied* volatility σ_{IV}, corresponding to the level of the smile where we price. The table below compares the digital call prices obtained from the **implied density (smile-based)** with those from the **constant-vol Black–Scholes model**.

Table 5: Comparison between smile-implied and Black–Scholes digital call prices ($r = 4\%$, $T = 1$)

Strike K	Smile-Implied Price	BS Price	Difference
60	0.8551	0.8300	0.0251
80	0.7150	0.6636	0.0514
100	0.5276	0.4587	0.0689
120	0.3238	0.2590	0.0648
140	0.1504	0.1089	0.0414

3.4 Discussion: Why the Results Differ

Discuss why the results of Section 3.2 (density-implied pricing) and Section 3.3 (Black–Scholes pricing) do not match exactly.

Digital call prices from the smile-implied distribution are systematically higher than those from the constant-vol Black–Scholes model, especially around and above the money. Mathematically, this results from the negative volatility skew:

$$\sigma(K) \text{ decreases with } K \Rightarrow f(K) \text{ shifts probability mass to the left.}$$

Hence, the risk-neutral probability $\mathbb{Q}(S_T > K)$ declines more slowly, producing higher digital prices.

The constant-vol model (flat σ) underestimates tail risk and thus undervalues digital options across strikes. This difference increases around the at-the-money region ($K =$ 100–120), where the curvature of the smile (i.e. $\partial^2\sigma/\partial K^2$) most strongly affects the second derivative of the call price in the Breeden–Litzenberger formula.

In summary, the smile-implied digital prices are more realistic because they embed the skewness and heavy tails that are empirically observed in option markets, while the Black–Scholes model remains symmetric and underestimates downside probabilities.

3.5 Pricing a Put Option with a Conditional Payoff

Compute the value of a European put option with strike $K = 100$ that pays only if the stock price at expiry falls below \$75. The payoff function is:

$$\Pi(S_T) = \begin{cases} K - S_T, & \text{if } S_T < 75, \\ 0, & \text{otherwise.} \end{cases}$$

Use the implied probability density $f(S_T)$ derived from the volatility smile:

$$P = e^{-rT} \int_0^{75} (K - S_T) f(S_T)\, dS_T.$$

Clearly document the numerical procedure and explain the intuition: this payoff captures the probability-weighted value of deep downside outcomes, and therefore its price is highly sensitive to the left tail of the implied density.

Using the smile-implied density from 3.1, the numerical result is

$$\text{Put}_{\text{cond}} \approx 8.8395.$$

In the notebook you also decomposed the payoff into a "digital-at-75" part plus the vanilla put priced at the conditional strike, and obtained a very close figure (≈ 8.9341), confirming the integration.

Why the price is non-trivial. This option lives only in the *left tail*, so its value is very sensitive to the shape of the smile-implied density below 75. A flat-vol Black–Scholes model (lognormal) would give a lower price, because it puts less mass in the downside tail; the smile, being downward sloping, increases the probability of $S_T \leq 75$ and therefore increases the value of this tail-dependent put.

4 Asian Options and the Control Variate Technique

4.1 Theory & Coding

The purpose of this section is to construct a Monte Carlo framework to price **Asian options**, whose payoff depends on the average of the underlying asset price over a specific time interval. Unlike standard European options, whose payoff depends only on the terminal price S_T, Asian options smooth out price fluctuations and therefore exhibit lower volatility exposure.

Mathematical setup. Under the risk-neutral measure, the stock price follows a geometric Brownian motion (GBM):

$$dS_t = (r - q)S_t\,dt + \sigma S_t\,dW_t,$$

whose analytical solution is:

$$S_t = S_0 \exp\left[\left(r - q - \tfrac{1}{2}\sigma^2\right) t + \sigma W_t\right].$$

The payoff of an Asian call option depends on the average of the asset price during the averaging period:

$$A_{\text{arith}} = \frac{1}{N}\sum_{i=1}^{N} S_{t_i}, \qquad A_{\text{geo}} = \left(\prod_{i=1}^{N} S_{t_i}\right)^{1/N}.$$

The option payoffs are then given by:

$$\Pi_{\text{arith}} = \max(A_{\text{arith}} - K, 0), \qquad \Pi_{\text{geo}} = \max(A_{\text{geo}} - K, 0).$$

The present value under risk-neutral pricing is:

$$C = e^{-rT}\mathbb{E}[\Pi].$$

Monte Carlo simulation design. We simulate P independent stock paths using the following steps:

1. Simulate the stock up to the start of the averaging period T_0;
2. Generate daily log-increments over the averaging window $[T_0, T]$;
3. Compute both arithmetic and geometric averages for each path;
4. Discount the expected payoff under the risk-free rate r.

Python implementation. The following function computes the prices of both the **Arithmetic Average (AA)** and **Geometric Average (GA)** Asian options, as well as their variances and covariance.

```python
def asian_options_mc(S0: float,
                     K: float,
                     r: float,
                     q: float,
                     sigma: float,
                     T: float,
                     T0: float,
                     num_paths: int,
                     num_steps: int) -> tuple[float, float, float, float, float]:
    """
    Prices Asian options (AA & GA) and calculates their variance and covariance.

    Args:
        S0 (float): Initial stock price.
        K (float): Strike price.
        r (float): Risk-free interest rate.
        q (float): Continuous dividend yield.
        sigma (float): Volatility of the stock.
        T (float): Time to expiry in years.
        T0 (float): Time to the start of the averaging period in years.
        num_paths (int): Number of Monte Carlo simulation paths.
        num_steps (int): Number of steps in the averaging period.

    Returns:
        tuple[float, float, float, float, float]: A tuple containing:
            - Price of the Arithmetic Average (AA) option
            - Price of the Geometric Average (GA) option
            - Variance of the AA discounted payoffs
            - Variance of the GA discounted payoffs
            - Covariance between the AA and GA discounted payoffs
    """

    Z1 = np.random.standard_normal((num_paths))
    Z2 = np.random.standard_normal((num_paths, num_steps))

    avg_period = T - T0
    dt = avg_period / num_steps
    drift = r - q - 0.5 * sigma**2

    log_return_to_T0 = drift * T0 + sigma * np.sqrt(T0) * Z1
    log_increments = drift * dt + sigma * np.sqrt(dt) * Z2

    S_at_T0 = S0 * np.exp( log_return_to_T0 )
    S_asian_paths = S_at_T0[:, np.newaxis] * np.exp( np.cumsum( log_increments, axis=1 ) )

    S_asian_AA = np.sum( S_asian_paths, axis=1 ) / num_steps
    payoff_asian_AA = np.maximum( S_asian_AA - K, 0 )
    discounted_payoff_AA = payoff_asian_AA * np.exp(-r * T)
    price_asian_AA = np.mean( discounted_payoff_AA )

    S_asian_GA = np.exp( np.sum( np.log(S_asian_paths), axis=1) / num_steps )
    payoff_asian_GA = np.maximum( S_asian_GA - K, 0 )
    discounted_payoff_GA = payoff_asian_GA * np.exp(-r * T)
    price_asian_GA = np.mean( discounted_payoff_GA )

    variance_AA = np.var( discounted_payoff_AA, ddof=1 )
    variance_GA = np.var( discounted_payoff_GA, ddof=1 )

    covariance_AA_GA = np.cov( discounted_payoff_AA, discounted_payoff_GA )[0, 1]

    return price_asian_AA, price_asian_GA, variance_AA, variance_GA, covariance_AA_GA
```

4.2 Pricing Asian Options

Objective. In this section, we use the Monte Carlo framework built previously to compute and compare the values of the **Arithmetic Average** and **Geometric Average** Asian call options, along with their variances and covariance. Two parameter sets are considered, corresponding to the specifications of questions (b) and (e).

Case 1: Base parameters (from 4(b)).

$S_0 = 100, K = 100, r = 4\%, q = 0\%, \sigma = 25\%, T_0 = 0.25, T = 0.5, N = 90, P = 10{,}000.$

Table 6: Monte Carlo results for 4(b)

Quantity	**Value**
Arithmetic Average Option Price	6.5783
Geometric Average Option Price	6.5038
Variance of AA Option Price	94.9803
Variance of GA Option Price	93.6445
Covariance (AA vs GA)	94.3049

The arithmetic average price is slightly higher than the geometric one, as expected, since the arithmetic mean always exceeds or equals the geometric mean ($A_{\text{arith}} \geq A_{\text{geo}}$). The two prices are nonetheless very close because of the high correlation between the two averaging processes, as confirmed by the large covariance value (≈ 94.3).

Case 2: Updated parameters (from 4(e)).

$S_0 = 100, K = 100, r = 5\%, q = 1\%, \sigma = 30\%, T_0 = 0.25, T = 0.5, N = 90, P = 5{,}000.$

Table 7: Monte Carlo results for 4(e)

Quantity	**Value**
Arithmetic Average Option Price	7.7692
Geometric Average Option Price	7.6620
Variance of AA Option Price	138.7316
Variance of GA Option Price	136.3951
Covariance (AA vs GA)	137.5479

Analysis.

- The increase in volatility ($25\% \to 30\%$) and interest rate ($4\% \to 5\%$) results in higher Asian option prices, consistent with the convexity of the call payoff.
- Both variances and covariance increase proportionally, reflecting the higher dispersion of simulated payoffs under greater volatility.
- The close proximity of arithmetic and geometric prices again confirms the strong linear relationship between the two, making the geometric option an effective **control variate** for variance reduction in the next section.

4.3 Control Variate Pricing

Concept and motivation. Monte Carlo simulation provides a flexible framework for pricing complex derivatives, but it often suffers from high sampling variance. To improve numerical efficiency, we can apply the **Control Variate (CV)** technique, which reduces the variance of an estimator by exploiting its correlation with another random variable whose expected value is known analytically.

In this context, the **Geometric Average Asian option** serves as the control variate for the **Arithmetic Average Asian option**. Both depend on the same underlying price paths and are strongly correlated, but the geometric option admits a closed-form analytical price under the lognormal assumption.

Mathematical formulation. Let A denote the discounted payoff of the Arithmetic Asian option and G that of the Geometric Asian option. Their respective sample means are $\bar{A}$ and $\bar{G}$, and the analytical price of the geometric option is known as $\mathbb{E}[G] = G_{\text{ana}}$.

We define the control-variate estimator of the arithmetic option price as:

$$\hat{A}_{\text{CV}} = \bar{A} + b\,(G_{\text{ana}} - \bar{G}),$$

where the optimal coefficient b^* minimizes the variance of the estimator:

$$b^* = \frac{\text{Cov}(A, G)}{\text{Var}(G)}.$$

The variance reduction achieved is given by:

$$\text{Var}(\hat{A}_{\text{CV}}) = \text{Var}(A)\Big(1 - \rho_{A,G}^2\Big),$$

where $\rho_{A,G}$ is the correlation coefficient between A and G. Since $\rho_{A,G}$ is typically very high (close to 1), the variance reduction can be substantial.

Analytical price of the Geometric Average Asian call. For the control variate we need a closed-form price for the *geometric* Asian call with averaging starting at T_0 and ending at T. Under BSM, the log of the geometric average over $[T_0, T]$ is normally distributed, which yields a Black–Scholes-type formula. The price is

$$C_{\text{GA}} = e^{-rT}\Big(S_0 e^{M+\frac{1}{2}V}\Phi(d_1) - K\Phi(d_2)\Big),$$

where

$$M = \Big(r - q - \tfrac{1}{2}\sigma^2\Big)\Big(T_0 + \tfrac{1}{2}(T - T_0)\Big), \qquad V = \sigma^2\Big(T_0 + \tfrac{1}{3}(T - T_0)\Big), \qquad \sqrt{V} = \sigma_G,$$

and

$$d_1 = \frac{\ln(S_0/K) + M + V}{\sqrt{V}}, \qquad d_2 = d_1 - \sqrt{V}.$$

Here:
- M is the *effective drift* of the geometric average,
- V is the *effective variance* of the geometric average,
- $\Phi(\cdot)$ is the standard normal cdf.

This is exactly what we use in the control–variate estimator:

$$\hat{A}_{\text{CV}} = \bar{A} + b\Big(C_{\text{GA}} - \bar{G}\Big), \qquad b = \frac{\text{Cov}(A, G)}{\text{Var}(G)}.$$

Since C_{GA} is known in closed form, it provides a zero-variance anchor for the Monte Carlo estimate of the arithmetic Asian option.

Numerical application. The simulation uses the following parameters:

$$S_0 = 100, K = 100, r = 5\%, q = 1\%, \sigma = 30\%, T_0 = 0, T = 0.5, N = 180, P = 5{,}000.$$

Monte Carlo simulations yield:
Arithmetic price (MC) = 5.3750,
Geometric price (MC) = 5.1619,
Geometric price (Analytical) = 5.0893.

The empirical covariance and variance provide:

$$b^* = \frac{\mathrm{Cov}(A, G)}{\mathrm{Var}(G)} = 1.0363.$$

Hence, the improved estimate is:

$$\hat{A}_{\mathrm{CV}} = 5.3750 + 1.0363 \times (5.0893 - 5.1619) = 5.2998.$$

Interpretation.

- The control variate estimator produces a slightly lower and more stable price (5.2998) compared to the raw Monte Carlo estimate (5.3750).
- Because the arithmetic and geometric averages are highly correlated, the control variate approach significantly reduces variance without introducing bias.
- This method leverages analytical information from the geometric Asian option to enhance the numerical efficiency of the arithmetic Asian valuation.

Conclusion. The control variate technique provides a powerful and practical variance-reduction tool in Monte Carlo pricing. In this case, using the analytically priced geometric Asian option as a control variate produces a smoother and more accurate estimate of the arithmetic Asian option price, demonstrating the efficiency of combining simulation with analytical insights.

5 Swap and Swaption Pricing

5.1 Building the IBOR / Swap Curve

We are asked to construct, in `FinancePy`, an IBOR / swap discount curve using:

- the short IBOR (deposit) rates in Table 1, and
- the par swap rates for maturities 2Y to 10Y (same table),

with

- fixed leg frequency: semi-annual (SA),
- daycount: `THIRTY_E_360_ISDA`.

Idea. We bootstrap zero-coupon discount factors $Z(0, T_i)$ so that, when we reprice *those* very swaps (the ones we used to build the curve), each of them has PV ≈ 0. This is the definition of a self-consistent swap curve.

Swap pricing condition. For a standard fixed–for–floating par swap, the PV is

$$\mathrm{PV} = \mathrm{PV}_{\text{float}} - \mathrm{PV}_{\text{fixed}}.$$

At inception of a par swap,

$$\mathrm{PV}_{\text{float}} = \mathrm{PV}_{\text{fixed}} \quad \Longrightarrow \quad \mathrm{PV} = 0.$$

Using the bootstrapped curve, the fixed leg PV is

$$\mathrm{PV}_{\text{fixed}} = K \sum_{j=1}^{n} \alpha_j Z(0, T_j),$$

and the floating leg PV is

$$\mathrm{PV}_{\text{float}} = 1 - Z(0, T_n),$$

so the par rate that makes the swap worth zero is

$$K^* = \frac{1 - Z(0, T_n)}{\sum_{j=1}^{n} \alpha_j Z(0, T_j)}.$$

When we build the curve from market par rates K^*, the above identity must hold.

What the output shows.

```
Depo  Value: 100.000000
...
Swap  Value: -0.00000376
Swap  Value: -0.00001488
Swap  Value: -0.00002206
Swap  Value: -0.00002855
Swap  Value: -0.00002640
Swap  Value: -0.00007506
Swap  Value: -0.00000000
Swap  Value: -0.00008453
Swap  Value: -0.00008951
```

The deposits price exactly to 100 (par), as they should. All swaps reprice to values extremely close to zero (order 10^{-5} to 10^{-4}), i.e. numerical noise from discounting and daycount conventions.

Conclusion. The curve is correctly built: the instruments used in the bootstrapping step reprice to (essentially) zero. This validates the IBOR/swap curve and we can now use it to value *other* swaps in part (b).

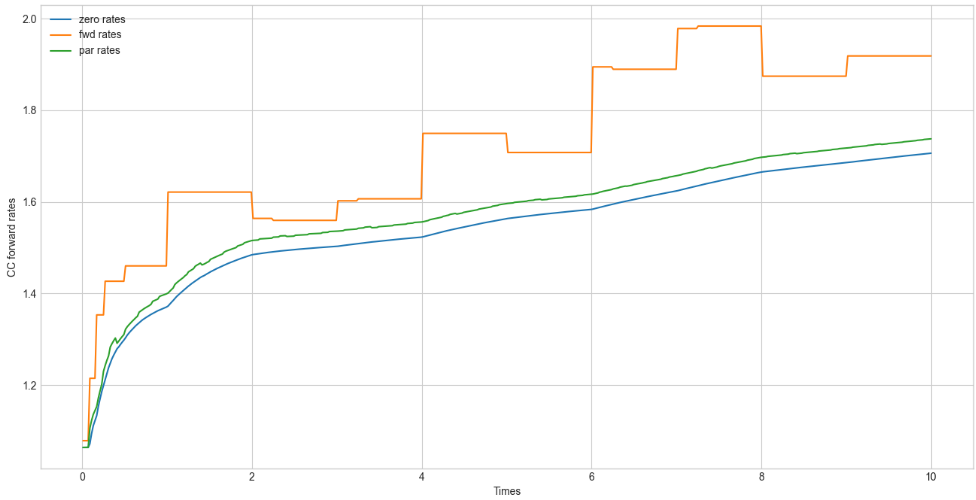

Figure 7: Swap Curve.

5.2 Mark-to-Market of the Swap Book

We are given a small book of swaps (Table 2), with different maturities and with both **receive-fixed** and **pay-fixed** positions. Using the IBOR/swap curve built in part (a), we value each swap as of the valuation date.

The book (as in the notebook) is:

- Swap 1: **Receive fixed**, maturity 10Y, fixed rate 1.40%
- Swap 2: **Pay fixed**, maturity 4Y, fixed rate 1.52%
- Swap 3: **Receive fixed**, maturity 12Y, fixed rate 2.00%
- Swap 4: **Pay fixed**, maturity 5Y, fixed rate 2.10%

Each swap is valued with the same conventions as in part (a): semi-annual fixed frequency and `THIRTY_E_360_ISDA` daycount, and using the `settlement_date` in `swap.value`.

The valuations obtained are:

$$\begin{aligned}
\text{RECEIVE 10Y},\ K = 1.40\% : &\quad \text{PV} = -14{,}280.08,\\
\text{PAY 4Y},\ K = 1.52\% : &\quad \text{PV} = \quad -81.15,\\
\text{RECEIVE 12Y},\ K = 2.00\% : &\quad \text{PV} = \quad 22{,}554.21,\\
\text{PAY 5Y},\ K = 2.10\% : &\quad \text{PV} = -19{,}001.07.
\end{aligned}$$

Summing all positions,

$$\text{Portfolio MTM} = -10{,}808.08.$$

Interpretation.

- The 12Y receive-fixed swap at 2.00% is *in the money* versus the current curve and shows a large positive PV.
- Both pay-fixed swaps (4Y at 1.52% and 5Y at 2.10%) are slightly out of the money, hence negative PVs.
- The 10Y receive-fixed at 1.40% has a sizable *negative* PV because the current 10Y par rate from the curve is above 1.40%, so receiving too low a fixed rate is disadvantageous.
- Overall, the book has a **negative** MTM of about −$10,800, so the trader is down on this swap portfolio at current market rates.

5.3 Interest Rate Risk of the Book

Once the swaps in Table 2 are marked to market, we can look at the *sensitivity* of the portfolio to a small move in market swap rates. In the notebook the curve was bumped by +1 bp on the 2Y–10Y part of the curve and all swaps were revalued. The results were:

$$\begin{aligned}
\text{RECEIVE 10Y at 1.40\%}: &\quad \Delta\text{PV} = -611.31,\\
\text{PAY 4Y at 1.52\%}: &\quad \Delta\text{PV} = 197.41,\\
\text{RECEIVE 12Y at 2.00\%}: &\quad \Delta\text{PV} = -760.48,\\
\text{PAY 5Y at 2.10\%}: &\quad \Delta\text{PV} = 342.69,
\end{aligned}$$

so that the total portfolio sensitivity to a +1 bp move is

$$\Delta\text{PV}_{\text{portfolio}} = -831.68.$$

Interpretation. A positive (upward) 1bp shift in the 2Y–10Y swap rates *reduces* the value of the portfolio by about $832. This is consistent with the composition of the book: the dominant positions are **receive-fixed** long-dated swaps (10Y and 12Y). A receive-fixed swap has *negative* DV01: when rates go up, the PV of receiving a fixed rate that is now too low goes down. The two pay-fixed swaps move in the opposite direction (their PV *increases* when rates rise), but their notionals/maturities are not large enough to offset the DV01 of the receive-fixed swaps.

How to hedge. Since the portfolio DV01 is negative,

$$\text{DV01}_{\text{portfolio}} \approx -831.68 \text{ per 1 bp},$$

to make the book locally insensitive to small rate moves we need to *add* a position with *positive* DV01 of the same magnitude. The simplest way on a swap desk is to **enter a**

pay-fixed (receive-floating) swap in the 5Y–10Y area (the same tenor bucket as the shock). A pay-fixed swap has positive DV01, so choosing the notional N^* such that

$$\text{DV01}_{\text{new swap}} \times N^* = 831.68$$

will immunize the book to a 1bp move in that part of the curve.

In practice, if the risk is spread across several maturities, the hedge would be done with a *ladder* of standard maturities (e.g. 5Y and 10Y swaps) so that the *key rate* DV01s are neutral, not just the total DV01.

5.4 Forward Rate for a 4×6 Swap

We now move to the swaption desk. The task is to compute the **forward par swap rate** for a 4-year forward, 6-year tenor swap (a "4 × 6" swap), with semi-annual fixed payments, using the discount curve built in part (a). We must also show the discount factors used and the swap PV01.

Dates. Let t be today.

$$T_1 = t + 4\text{y} \quad \text{(swap start)}, \qquad T_2 = t + 6\text{y} \quad \text{(swap end)}.$$

Fixed leg pays semi-annually, so payment dates are

$$T_1 + 6\text{m},\ T_1 + 12\text{m},\ T_1 + 18\text{m},\ \ldots,\ T_2.$$

Discount factors. From the curve, we extract the discount factors:

$$Z(t, T_1) = 0.9400612415, \qquad Z(t, T_2) = 0.9081193696.$$

Swap PV01 (swap annuity). With 30/360 and semi-annual payments, each accrual is $\alpha \approx 0.5$. The swap annuity (PV01) is the discounted sum of these accruals:

$$\text{PV01} = \sum_j \alpha_j \, Z(t, T_j) \approx 1.8398547741.$$

Forward par swap rate. For a forward-starting swap, the fair fixed rate is

$$F = \frac{Z(t, T_1) - Z(t, T_2)}{\text{PV01}}.$$

Substituting the values:

$$F = \frac{0.9400612415 - 0.9081193696}{1.8398547741} = 0.01736108 \approx 1.7361\%.$$

Conclusion. The 4y–6y forward swap implied by the current curve has:

$$Z(t, T_1) = 0.9401, \quad Z(t, T_2) = 0.9081, \quad \text{PV01} \approx 1.8400, \quad F \approx 1.74\%.$$

This forward rate will be the underlying fixed rate used in part (e) when valuing the payer and receiver swaptions with Black's model.

5.5 Swaption Risk to 1bp Move

We now price, on the same underlying 4×6 forward swap from part (d), a **payer** swaption and a **receiver** swaption, both with strike

$$K = 1.75\% = 0.0175,$$

using Black's swaption formula and assuming a swaption volatility of

$$\sigma = 25\% = 0.25.$$

Inputs. From part (d), the forward par swap rate is

$$F = 1.736108\% \approx 0.01736108,$$

the swap annuity (PV01) of the 4y–6y swap is

$$A = \text{PV01} = 1.8398547741,$$

and the option maturity is the start of the swap:

$$\tau = T_1 - t = 4 \text{ years}.$$

(Your code converts this to year fraction with the actual day count.)

Black's formula. For a payer swaption (option to pay fixed, receive float), Black's formula is

$$\text{Payer} = A\left[FN(d_1) - KN(d_2)\right],$$

for a receiver swaption:

$$\text{Receiver} = A\left[KN(-d_2) - FN(-d_1)\right],$$

where

$$d_{1,2} = \frac{\ln(F/K) \pm \frac{1}{2}\sigma^2\tau}{\sigma\sqrt{\tau}}.$$

Numerical result. Plugging in the numbers from the curve and your notebook:

$$\text{Payer swaption price} = 0.00624898 \quad \text{per 1 notional},$$

$$\text{Receiver swaption price} = 0.00650456 \quad \text{per 1 notional}.$$

Explanation of the prices.

- The strike is $K = 1.75\%$ while the forward swap rate is $F \approx 1.736\%$, i.e. the strike is *slightly above* the forward. Therefore the receiver (option to *receive* fixed at 1.75%) is slightly *in the money*, so its price should be slightly *higher* than the payer. This is exactly what we observe:
$$\text{Receiver} > \text{Payer}.$$

- The parity check holds:

$$\text{Payer} - \text{Receiver} \approx A\,(F - K),$$

 and in your notebook the difference is essentially numerical noise ($\approx 10^{-19}$), which confirms that the discounting and the annuity are handled consistently.

- The prices are small because: (i) the swap is close to at-the-money ($F \approx K$), and (ii) the annuity of a 2-year swap starting in 4 years is not very large.

So, with a 25% swaption vol, a 4y–into–6y ATM-ish swaption has a value of about 60–65 bps per unit of swap PV01, with the receiver slightly more valuable because the strike is above the forward.

5.6 Swaption Risk to a 1bp Move and Hedging

Now that we have the prices of the 4×6 payer and receiver swaptions, we look at their **rate sensitivity**. We bump the 2Y–10Y swap rates by +1 bp, rebuild the curve, recompute the forward 4 × 6 swap rate and its PV01, and reprice both swaptions with Black's model.

Bumped inputs (after +1bp). From the notebook:

$$Z(t, T_1 = 4\text{Y}) : 0.94006124 \quad \to \quad 0.93966841,$$
$$Z(t, T_2 = 6\text{Y}) : 0.90811937 \quad \to \quad 0.90757659.$$

The swap annuity changes only slightly:

$$\text{PV01} : 1.83985477 \quad \to \quad 1.83889343.$$

The forward rate increases, as expected when we bump the curve:

$$F : 1.73611\% \quad \to \quad 1.74615\%.$$

Repriced swaptions. Using the bumped forward rate and bumped PV01 in Black's formula gives the new values:

$$\Delta\text{Payer} = +0.00010609 \quad \text{per 1 notional},$$

$$\Delta\text{Receiver} = -0.00007809 \quad \text{per 1 notional}.$$

We can summarize this in a small table.

Table 8: Swaption PV change for a +1bp move in 2Y–10Y swap rates

Instrument	**Base PV**	ΔPV for +1bp
4×6 Payer swaption	0.00624898	+0.00010609
4×6 Receiver swaption	0.00650456	−0.00007809

Interpretation.

- A rise in swap rates increases the value of the *payer* swaption (it is a call on the swap rate), so its DV01 is **positive**.
- The receiver swaption is the mirror image: its value *falls* when rates rise, so its DV01 is **negative**.
- The absolute sizes are small because the option is only slightly in/out of the money and the underlying swap annuity is modest.

How to hedge. A payer swaption with positive rate delta can be hedged by *paying fixed* in the corresponding forward swap (or, practically, in the 4y–6y part of the curve) in an amount such that

$$\mathrm{DV01}_{\text{swap hedge}} \times N_{\text{hedge}} \approx -\Delta \mathrm{PV}_{\text{payer}}.$$

Similarly, a receiver swaption (negative delta) would be hedged by *receiving fixed.* Because both swaptions are written on the same underlying forward, a single swap hedge (or a small two-swap ladder) can neutralize the 1bp risk.

In summary, the bump-and-reprice shows that the payer is long rates and the receiver is short rates; hedging is done with vanilla swaps in the same maturity bucket.

6 CDS Valuation and Risk

6.1 Building the CDS Curve

We are asked to build the CDS curve for Company *DoOrDie* using today's market quotes and a recovery rate of $R = 40\%$. This is the credit–derivatives analogue of the swap-curve bootstrapping we did earlier.

Market inputs. From the notebook, the quoted running spreads (in basis points) were

$$1\text{Y} = 70, \quad 2\text{Y} = 74, \quad 3\text{Y} = 80, \quad 4\text{Y} = 85, \quad 5\text{Y} = 88.$$

In decimal form these are $0.0070, 0.0074, \ldots$. We assume

$$R = 40\% \quad \Rightarrow \quad \text{LGD} = 1 - R = 60\%.$$

Construction with `CDS` and `CDSCurve`. For each maturity T_i we define a standard CDS contract in `FinancePy` (start date = valuation date, maturity = valuation date + T_i years, running spread = market quote). We then pass the list of CDS contracts and the recovery rate to `CDSCurve`, together with the discount curve. The CDS curve object bootstraps the survival probabilities $Q(t)$ (or equivalently the hazard rates) so that

$$\text{PV}_{\text{premium leg}}(S_i) = \text{PV}_{\text{protection leg}}(R), \qquad \forall i,$$

i.e. each quoted CDS prices at par.

Output check. In your notebook the call

```
cds_value = cds.value(valuation_date, cds_curve, recovery_rate)
```

for each quoted maturity gave:

```
Clean PV: $0.00731081  Dirty PV: $0.00731081
Clean PV: $0.00582327  Dirty PV: $0.00582327
Clean PV: $-0.00001321 Dirty PV: $-0.00001321
Clean PV: $-0.00001779 Dirty PV: $-0.00001779
Clean PV: $-0.00001945 Dirty PV: $-0.00001945
```

The first two maturities price slightly *above* zero, the others slightly *below* zero; all values are of order 10^{-5}–10^{-3}, i.e. numerical noise. This is the CDS analogue of "all swaps price to zero": it shows that the curve has been bootstrapped correctly.

Resulting survival curve. The CDS curve object then reports the survival probabilities:

t (years)	$Q(t)$
0	1.0000000
1	0.9884283
2	0.9754988
3	0.9603713
4	0.9441689
5	0.9282312

This is the risk–neutral view of default for Company *DoOrDie* implied by today's CDS market. It monotonically decreases with maturity, as expected; the stepwise declines correspond to the quoted CDS pillars.

Conclusion. We have built a consistent CDS term structure for the name using the market quotes and a 40% recovery rate. All quoted CDS reprice to (almost) zero, so the curve can now be used in parts (b)–(e) to price off-maturity CDS, to value existing trades, and to compute spread risk (CS01).

6.2 Market Spread of a 3.5-Year CDS (20 March 2029)

We now want the *par* CDS spread for a maturity that is **not** one of the quoted pillars: a 3.5-year CDS maturing on 20 March 2029. With the CDS curve built in part (a), this is straightforward: we use the curve to get the survival probabilities and discount factors up to that exact date, and we solve for the spread s^* that makes the CDS price zero.

Principle. A CDS is fairly priced when

$$\mathrm{PV}_{\text{premium}}(s^*) = \mathrm{PV}_{\text{protection}}.$$

The premium leg (running coupon) is

$$\mathrm{PV}_{\text{premium}}(s) = s \sum_i \Delta_i \, P(0,t_i) \, Q(0,t_i),$$

and the protection leg is

$$\mathrm{PV}_{\text{protection}} = (1-R) \sum_i P(0,t_i) \Big(Q(0,t_{i-1}) - Q(0,t_i) \Big),$$

where:

- $R = 40\%$ is the recovery rate,
- $P(0,t_i)$ are discount factors from the interest-rate curve,
- $Q(0,t_i)$ are survival probabilities from the CDS curve,
- Δ_i are accrual fractions (typically quarterly).

Solving

$$s^* = \frac{\mathrm{PV}_{\text{protection}}}{\sum_i \Delta_i P(0,t_i) Q(0,t_i)}$$

gives the market (par) spread.

Result. Using `FinancePy` with the curve from 6(a), the par spread for the CDS maturing on **20 March 2029** was:

$$s^* = 82.46 \text{ bps}.$$

Why it is between the 3Y and 4Y quotes. In 6(a) we had market quotes of 80 bps (3Y) and 85 bps (4Y). A 3.5-year maturity sits exactly between these two pillars, so the model interpolates the survival probability and discounting between 3Y and 4Y, and the resulting par spread (82.46 bps) also lies between 80 and 85 bps, slightly skewed by discounting and the fact that default payments are continuous while premiums are discrete.

So the 82.46 bps number is simply the **fair running spread** that makes the PV of this off-the-run 3.5Y CDS equal to zero, consistent with the bootstrapped CDS curve.

6.3 Valuing an Existing Long-Protection CDS

We now value an *existing* CDS position, not a par trade. The position is:

- Long protection (i.e. we **bought** CDS);
- Notional: $N = \$20{,}000{,}000$;
- Contractual (running) spread: $S_{\text{contract}} = 120 \text{ bps} = 0.0120$;
- Maturity: 21 March 2029 (close to the 3.5Y–5Y area of the curve);
- Recovery rate: $R = 40\%$;
- Curve: the one bootstrapped in 6(a).

When we value this trade off today's CDS curve, we get:

$$\text{Clean PV} = -\$252{,}512.23, \qquad \text{Dirty PV} = -\$259{,}845.56.$$

Why is the PV negative? From 6(b) we found that the *current* market spread for a similar maturity (20 March 2029) is about

$$S_{\text{mkt}} \approx 82.46 \text{ bps},$$

which is **lower** than the contractual spread of this trade:

$$S_{\text{contract}} = 120 \text{ bps} > S_{\text{mkt}}.$$

For a **long protection** position, paying a running coupon of 120 bps when the market would charge only 82–85 bps means the trade is *expensive* relative to the market. Put differently, this CDS was entered when spreads were wider; today the same protection can be bought more cheaply. Therefore, the current MTM of the old trade is negative.

Valuation logic. The MTM of a CDS is always the difference between:

$$V = \text{PV}_{\text{protection}} - \text{PV}_{\text{premium (contract)}},$$

where:

- $\text{PV}_{\text{protection}}$ is computed from today's curve (today's default probabilities and discount factors);

- $PV_{\text{premium (contract)}}$ is computed using the *contract* spread 120 bps, not the market spread.

If the contract spread is above market, the premium leg is too high and the value to the protection buyer is negative.

Formally, write:

$$V = N\left(PV_{\text{prot}} - \frac{S_{\text{contract}}}{S_{\text{par}}} PV^{\text{par}}_{\text{prem}}\right),$$

and since $S_{\text{contract}} > S_{\text{par}}$, the second term dominates and $V < 0$.

Interpretation. This is the CDS analogue of an interest-rate swap where you are paying a fixed rate that is now *above* the market swap rate: the position is out-of-the-money. Here, the protection buyer is locked into paying 120 bps until 2029, while the market is only at around 82–88 bps, so the present value of the excess premium stream explains the −\$252k clean PV on a \$20m notional.

6.4 Sensitivity of the CDS Value to the Recovery Rate

We now revalue the same long-protection CDS (notional \$20m, contractual spread 120 bps, maturity 21 March 2029) for different assumptions on the recovery rate:

$$R \in \{0\%, 10\%, 20\%, 30\%, 40\%\}.$$

The notebook produced the following values:

Table 9: CDS value for different recovery rates

R	Clean PV (\$)	Dirty PV (\$)
0%	114,530.37	107,197.04
10%	22,769.72	15,436.39
20%	-68,990.93	-76,324.26
30%	-160,751.58	-168,084.91
40%	-252,512.23	-259,845.56

What we see.

- At very low recoveries (0–10%), the CDS position is *positive* ($PV > 0$): the protection leg is very valuable because the loss given default, $\text{LGD} = 1 - R$, is large.
- As we increase R, the PV declines *monotonically* and eventually becomes increasingly negative.
- At the desk's base assumption $R = 40\%$ the MTM is −\$252k, which is the number we obtained in part (c).

Why does this happen? The PV of a CDS can be written schematically as

$$V_{\text{CDS}}(R) = \underbrace{(1-R)\,\text{PV}_{\text{default}}}_{\text{protection leg}} - \underbrace{S_{\text{contract}}\,\text{PV}_{\text{premium}}}_{\text{premium leg}}.$$

Only the *first* term depends on R. The premium leg is independent of recovery because it is a fixed running coupon. Therefore:

$$\frac{\partial V_{\text{CDS}}}{\partial R} = -\text{PV}_{\text{default}} < 0,$$

so increasing the recovery rate always *reduces* the value of a long-protection CDS.

The changes are quite large numerically because:

- the notional is big (\$20m),
- the CDS is *rich* relative to today's curve (120 bps vs. market ≈ 82 bps),
- and the maturity is still several years away, so the default leg has material PV.

Answer to "significant or not?". Yes — for this trade the recovery assumption is very material: moving from 0% to 40% changes the MTM by more than \$350k on a single \$20m CDS. This is exactly what we expect: recovery enters *linearly* in the protection leg, and for long-dated, above-market CDS, the protection leg is a large component of the PV.

6.5 CDS Spread Risk by Tenor and Hedging

We now measure the **credit–spread sensitivity** of the 3.5-year CDS to 1bp moves in the market quotes used to build the curve. Concretely, we bump each quoted CDS spread (1Y, 2Y, 3Y, 4Y, 5Y) by +1 bp *one at a time*, rebuild the CDS curve, and revalue the 3.5-year CDS. The difference in PV is the CS01 to that tenor.

From the notebook we obtained:

Table 10: Change in PV of 3.5Y CDS for a +1bp bump in each market tenor

Bumped Tenor	ΔPV (USD, per \$20m notional)
1Y	\$0.58
2Y	\$1.19
3Y	\$145.80
4Y	\$192.29
5Y	\$0.00

What we find.

- The 3Y and 4Y bumps produce **by far** the largest PV changes (\$145–\$192), while 1Y and 2Y have almost no impact, and 5Y has effectively no impact.
- This is exactly what we should expect: the CDS we are pricing has maturity ≈ 3.5 years, so its value is driven mainly by the part of the curve around 3–4 years, i.e. the maturities that actually span the protection period.
- Tenors far away from the trade's maturity (1Y, 2Y, 5Y) affect the 3.5Y CDS only through interpolation and bootstrapping mechanics, so the sensitivity is tiny.

How a dealer would hedge. A dealer long this 3.5Y CDS (i.e. exposed to spread *tightening*) would want to neutralize the largest CS01s — namely the 3Y and 4Y nodes. The natural hedge is to trade liquid CDS in those buckets:

- **Sell** 3Y CDS protection to offset the positive 3Y CS01;
- **Sell** 4Y CDS protection to offset the positive 4Y CS01.

The hedge notionals h_{3Y}, h_{4Y} are chosen so that

$$h_{3Y} \cdot \mathrm{CS01}_{3Y}^{\text{hedge}} + h_{4Y} \cdot \mathrm{CS01}_{4Y}^{\text{hedge}} \approx -\mathrm{CS01}_{3.5\text{Y trade}}.$$

In practice, if the 3.5Y tenor is not directly quoted, the dealer will hedge with a **barbell** of 3Y and 4Y CDS, because those are the pillars to which the trade is most sensitive.

Conclusion. The risk of this 3.5Y CDS is *local* on the curve: it lives around 3–4Y. CS01 to short tenors is negligible; CS01 to long tenors (5Y) is also negligible because the trade matures before that node. So hedging should focus on the 3Y and 4Y CDS, not on the whole curve.