Risk Analytics

Machine Learning and Optimization
for Data-Driven Decision Making

Fernando S. Oliveira

Draft version — June 5, 2026

Chapter 11
Tree-Based Dynamic Risk Estimation

Chapter 11

Tree-Based Dynamic Risk Estimation

11.1 Introduction: From Time Series to Conditional Risk Trees

Chapter 10 introduced time series as tools for monitoring variables that evolve through time. For many risk problems, however, monitoring is not enough. A risk manager does not only need to know that copper prices have been volatile. The manager needs to know what the remaining procurement loss could be if the price moves to a new level next month, and how that risk changes over the rest of the planning horizon.

A rolling-window estimate gives one current risk number from recent historical data. This is useful for surveillance, but it is not a conditional map of future risk. Chapter 11 develops that map using a recombining price lattice. The lattice converts historical price behaviour into future price nodes. After a procurement loss function is attached to the nodes, each node generates a conditional loss table. Expected loss, VaR, and CVaR are then computed from the losses still reachable from that node.

The first modelling question is whether the price level should be treated as stationary. If a price fluctuates around a stable mean, a pullback model may be appropriate. If the level is persistent or non-stationary, imposing pullback can understate risk. This chapter therefore starts with simple diagnostics for non-stationarity. The copper application is treated as a persistent price-level process: the level is not assumed to return quickly to a fixed mean, while monthly log price changes are used to calibrate the scale of future lattice movements.

The chapter bridges three earlier parts of the book. Chapters 4–6 developed loss distributions, VaR, CVaR, and scenario-based tail risk. Chapter 10 showed how risk drivers evolve over time. This chapter combines both ideas: a price lattice turns a time series into conditional future loss tables. The

© 2026 Fernando S. Oliveira

objective is transparency: the analyst should be able to state the data used, the non-stationarity diagnostics, the lattice calibration, the loss function, and the tail-risk calculation at each node.

The application is commodity procurement risk. Commodity prices are not generic financial time series. Their dynamics depend on storability, inventories, production adjustment, seasonality, and the distinction between spot and forward markets [10, 7]. Copper is used as the running case because it is a major input in manufacturing, construction, electrification, infrastructure, and energy-transition supply chains.

The chapter proceeds as follows. Section 11.2 uses simple diagnostics to assess non-stationarity. Section 11.3 builds a random-walk price lattice from historical price changes. Section 11.4 shows how backward induction creates conditional loss tables. Section 11.5 computes VaR and CVaR from those tables and applies the method to the copper procurement example. The chapter ends with practical conclusions for risk analysts.

11.2 Testing for Non-Stationarity with Simple Diagnostics

Before building a price lattice, the analyst should decide whether the price level can reasonably be treated as stable around a fixed mean. This is not a purely statistical question. It affects the risk model. If the price level is persistent, the lattice should allow shocks to remain in the price level. If the price level is stationary, a model with pullback toward a reference level may be more appropriate. This chapter uses simple diagnostics rather than a full econometric modelling exercise.

The first diagnostic is visual. A stationary price level should fluctuate around a relatively stable centre. A non-stationary price level may drift, move across regimes, or remain far from earlier historical levels for long periods. Figure 11.1 plots monthly copper prices and a 36-month rolling mean. The rolling mean changes materially over the sample, which is a warning against treating one historical average as a reliable short-run anchor.

© 2026 Fernando S. Oliveira

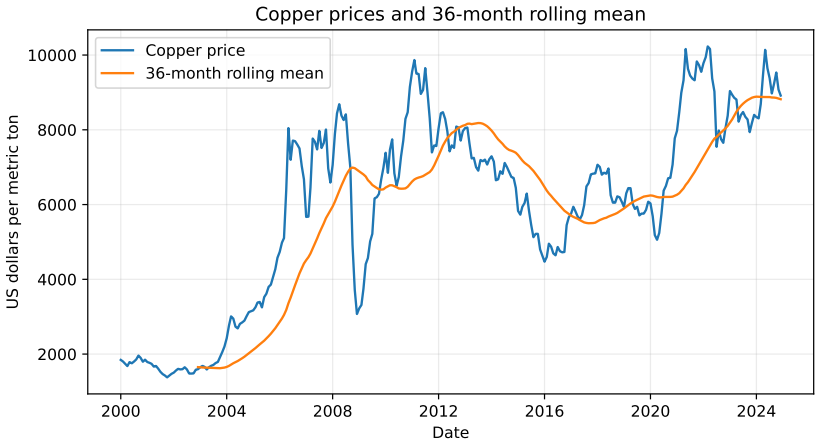

Figure 11.1: Copper prices and a 36-month rolling mean.

The second diagnostic is the autocorrelation function. For a series y_t, the lag-h autocorrelation measures the association between y_t and y_{t-h}. Lag zero is omitted because it is always equal to one and has no diagnostic value. A stationary series typically has autocorrelations that decline relatively quickly. A persistent or non-stationary price level often has autocorrelations that remain high over many lags. By contrast, price changes may show much weaker autocorrelation [1, 6].

© 2026 Fernando S. Oliveira

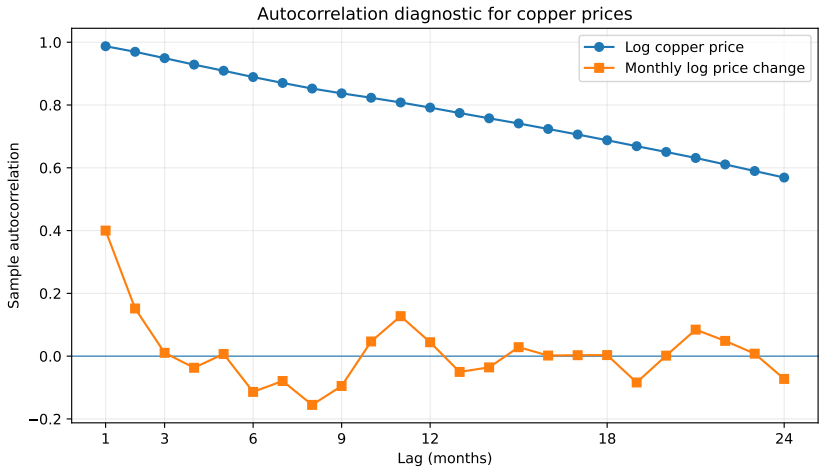

Figure 11.2: Autocorrelation of copper log prices and monthly log price changes, excluding lag zero.

Figure 11.2 shows a high and slowly declining autocorrelation for the log price level. Monthly log price changes have much weaker autocorrelation. This is the empirical pattern expected when the level is persistent but the increments are more stable.

The third diagnostic is to compare subperiods. Table 11.1 shows that average copper prices in 2000–2007 were far below the averages observed after 2008. This does not prove a specific stochastic process, but it reinforces the point that the price level has not behaved as a stable constant over the full sample.

Table 11.1: Copper price levels across subperiods

Sample	Mean price	Minimum price	Maximum price
2000–2007	3389	1377	8046
2008–2014	7232	3072	9868
2015–2024	7103	4472	10231

The fourth diagnostic is the augmented Dickey–Fuller (ADF) test. The ADF test is a standard unit-root test used to distinguish a persistent non-stationary level from a stationary process [4, 6]. For $y_t = \log P_t$, a simplified

© 2026 Fernando S. Oliveira

ADF regression is

$$\Delta y_t = a + b y_{t-1} + \sum_{i=1}^{q} c_i \Delta y_{t-i} + e_t. \tag{11.1}$$

The coefficient b is the key parameter. If $b = 0$, the level behaves like a unit-root process: shocks to the level are persistent. If $b < 0$ and statistically different from zero, the process tends to pull back after moving away from its previous level, which is evidence against a unit root. The ADF statistic is the test statistic for $b = 0$, using non-standard critical values. The p-value reports whether the unit-root null can be rejected. A small p-value supports stationarity; a large p-value means that the data do not provide enough evidence to reject a unit root.

Table 11.2: ADF diagnostics for copper log prices and monthly log price changes

Series tested	ADF statistic	p-value	Interpretation
Log price level, 2000–2024	-1.91	0.328	Unit root not rejected
Monthly log price changes, 2000–2024	-4.89	< 0.001	Unit root rejected

Table 11.2 gives the key diagnostic result. For log price levels, the test does not reject $b = 0$. The implication is not that the price must literally follow a random walk, but that the data do not justify imposing a stable short-run mean. For monthly log price changes, the test rejects the unit-root null. This supports the modelling strategy used in the chapter: treat the price level as persistent and calibrate the lattice from the more stable distribution of monthly log price changes.

These diagnostics lead to a disciplined modelling choice. The copper case below uses a non-stationary random-walk lattice. Shocks are allowed to persist in the price level, and no short-run pullback toward a fixed reference price is imposed. Mean-reverting commodity models may be appropriate in other settings, especially where storage, seasonality, production response, or physical balancing create a defensible short-run reference level. They are not used in this chapter.

11.3 Building Price Lattices from Historical Data

Let P_t denote the commodity price at time t. In the running case, P_t is the monthly copper price in US dollars per metric ton. The analyst starts

by measuring the volatility of monthly log price changes, $\log(P_t/P_{t-1})$. Log changes describe proportional movements and aggregate naturally over time.

A binomial price lattice starts from the current price P_0. At each future period, the price moves up or down:

$$P_{t+1} = \begin{cases} uP_t, & \text{with probability } p, \\ dP_t, & \text{with probability } 1-p, \end{cases}$$

where $u > 1$ is the upward multiplier and $0 < d < 1$ is the downward multiplier. After t periods, if k upward moves have occurred, the price at node (t, k) is

$$P_{t,k} = P_0 u^k d^{t-k}, \qquad k = 0, 1, \ldots, t.$$

The lattice is recombining because an up-then-down path reaches the same node as a down-then-up path:

$$P_0 ud = P_0 du.$$

This makes the structure compact. A full binary tree has 2^t paths after t periods, while a recombining binomial lattice has only $t+1$ distinct price nodes at that date. This is why binomial trees and lattices are widely used as discrete approximations to dynamic price uncertainty [3, 7, 5].

The multipliers u and d should reproduce the scale of one-period price uncertainty. A common choice, used in binomial approximations to lognormal price dynamics, is to set the up and down movements symmetrically in log-price space [3, 7]. If $\hat{\sigma}$ is the sample standard deviation of monthly log price changes, a one-period specification is

$$u = e^{\hat{\sigma}}, \qquad d = e^{-\hat{\sigma}}.$$

This means that one up move increases the log price by one estimated monthly standard deviation and one down move decreases it by the same amount. The product $ud = 1$ makes the movements symmetric around the current log price, so an up move followed by a down move returns to the same price node. For a time step of length Δt, the usual scaled version is

$$u = e^{\hat{\sigma}\sqrt{\Delta t}}, \qquad d = e^{-\hat{\sigma}\sqrt{\Delta t}},$$

because diffusion-type price uncertainty scales with the square root of time [7, 5]. In this chapter, the time step is one month, so $\Delta t = 1$ when $\hat{\sigma}$ is estimated from monthly data.

© 2026 Fernando S. Oliveira

The probability p must be interpreted carefully. In option pricing, lattices are often evaluated under risk-neutral probabilities. In this chapter, the objective is different: the analyst wants a real-world distribution of future procurement losses. The probability p is therefore a physical probability used for risk estimation, not a risk-neutral pricing probability. This distinction matters in commodity applications because a model may fit futures prices while still implying different real-world spot-price dynamics [2].

A simple empirical estimate is the historical frequency of positive monthly log price changes,

$$\hat{p} = \frac{1}{T_{\text{hist}}} \sum_{t=1}^{T_{\text{hist}}} \mathbf{1}\left\{\log(P_t/P_{t-1}) > 0\right\}.$$

Alternatively, p may be selected to match an estimated one-period expected growth factor. If

$$g = E\left[\frac{P_{t+1}}{P_t}\right],$$

then the lattice implies

$$g = pu + (1-p)d,$$

and therefore

$$p = \frac{g-d}{u-d}.$$

The calibration choice should always be reported: estimation window, price change definition, volatility estimate, drift assumption, and transition probability.

© 2026 Fernando S. Oliveira

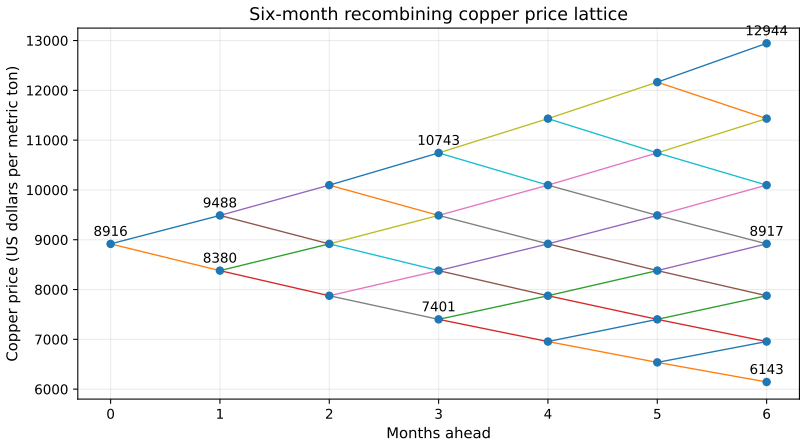

Figure 11.3: A six-month recombining copper price lattice calibrated from monthly copper price changes.

Worked Example 11.1: Calibrating the first copper-price nodes. Using monthly copper prices from January 2000 to December 2024, the monthly log-price-change volatility is 6.21%. A simple one-month lattice gives

$$u = e^{0.0621} = 1.064, \qquad d = e^{-0.0621} = 0.940.$$

Starting from the December 2024 copper price $P_0 = 8916$, the first two price nodes are approximately

$$P_{1,1} = 8916 \times 1.064 = 9488,$$

and

$$P_{1,0} = 8916 \times 0.940 = 8379.$$

The same sample has positive monthly log price changes in 54.85% of months, giving $\hat{p} = 0.549$.

11.4 Backward Induction

The previous sections built the future price lattice. Backward induction is the method that turns that lattice into a dynamic risk estimate. The logic is simple: first define the loss at the terminal nodes, then work backward to compute what each earlier node implies for future losses. In decision models, backward induction is used to choose actions. In this chapter, there are no

decisions yet. The method is used only to evaluate risk.

Backward induction starts with a loss function. In the copper procurement case, the firm is exposed to price increases relative to today's price. For a planned purchase of Q metric tons at horizon T, define the terminal downside loss as

$$L_{T,k} = Q \max(P_{T,k} - P_0, 0). \tag{11.2}$$

This loss is zero if the future price is below the current price, because the firm does not suffer an additional procurement cost. It is positive if the future price exceeds the current price. For results per metric ton, set $Q = 1$. For a larger purchase, multiply the per-ton losses by Q. Equation (11.2) is deliberately narrow: it isolates commodity price-increase risk before adding hedging, inventory, supplier, or production decisions.

Worked Example 11.2: Procurement loss function.
Suppose today's copper price is $P_0 = 8916$ dollars per metric ton and the firm plans to buy one metric ton in six months. If a terminal node has price $P_{6,k} = 12943$, then Equation (11.2) gives

$$L_{6,k} = \max(12943 - 8916, 0) = 4027.$$

If another terminal node has price $P_{6,k} = 8379$, the loss is

$$L_{6,k} = \max(8379 - 8916, 0) = 0.$$

The loss function measures downside procurement exposure, not total purchasing cost. It answers the question: how much more would the firm pay if it waits and the price is higher at the purchasing date?

For expected loss, backward induction is scalar. At terminal nodes,

$$V_{T,k} = L_{T,k}.$$

At an earlier node (t, k),

$$V_{t,k} = pV_{t+1,k+1} + (1 - p)V_{t+1,k}. \tag{11.3}$$

The value $V_{t,k}$ is the expected future loss conditional on being at time t and node k. It is not a price and it is not a decision value. It is a risk estimate: the average terminal downside loss implied by all future paths that can still be reached from that node.

Expected loss is not enough for downside-risk management. VaR and CVaR depend on the tail of the conditional loss distribution. Therefore,

backward induction for VaR and CVaR should propagate loss tables, not only scalar expected values.

At node (t, k), let $h = T - t$ be the number of periods remaining. If j of those h remaining periods are upward moves, the terminal price reachable from node (t, k) is

$$P_{t,k}^{T}(j) = P_{t,k} u^{j} d^{h-j}, \qquad j = 0, 1, \ldots, h. \tag{11.4}$$

The corresponding terminal loss is

$$\ell_{j}^{(t,k)} = Q \max(P_{t,k}^{T}(j) - P_0, 0). \tag{11.5}$$

For a constant-probability binomial lattice, the conditional probability of that terminal node is

$$\pi_{j}^{(t,k)} = \binom{h}{j} p^{j} (1-p)^{h-j}. \tag{11.6}$$

The conditional loss table at node (t, k) is obtained by listing the losses $\ell_{j}^{(t,k)}$ and their probabilities $\pi_{j}^{(t,k)}$. If several values of j produce the same loss, their probabilities are added. This aggregation is important for procurement losses because all terminal prices below P_0 produce the same loss, zero.

The distinction is important. Expected loss can be propagated as a single number because expectation is linear. VaR and CVaR cannot generally be obtained by averaging the VaR or CVaR of the child nodes. The tail of the combined loss table may be different from the weighted average of child-node tails. This is why VaR and CVaR are computed from the full conditional loss table [13, 12].

© 2026 Fernando S. Oliveira

Worked Example 11.3: Conditional loss table at a node.
Consider a two-period lattice from node (t, k). Suppose the upward probability is $p = 0.550$ at each step and the terminal losses after two steps are

$$\ell_0^{(t,k)} = 0, \qquad \ell_1^{(t,k)} = 300, \qquad \ell_2^{(t,k)} = 1200.$$

The middle loss receives probability from two paths, up-then-down and down-then-up. Using Equation (11.6), the conditional loss table is

Loss	0	300	1200
Probability	$(0.450)^2$	$2(0.550)(0.450)$	$(0.550)^2$

or approximately

Loss	0	300	1200
Probability	0.203	0.495	0.303.

This table, not the individual paths alone, is the object used to compute VaR and CVaR at the node.

11.5 Node-Conditional VaR, CVaR, and the Copper Risk Map

A price lattice becomes a risk model once each node has a conditional loss table. At time t and node k, the node-conditional VaR at confidence level α is

$$\mathrm{VaR}_{\alpha,t,k} = \inf\left\{\ell : P(L_{t:T} \le \ell \mid t, k) \ge \alpha\right\}.$$

It is the loss threshold that is exceeded with probability at most $1 - \alpha$, conditional on being at that node.

For CVaR, the most useful definition in a finite lattice is the Rockafellar–Uryasev representation. Suppose the conditional loss table at node (t, k) contains losses ℓ_m with probabilities π_m, where $m = 1, \ldots, M$ and $\sum_m \pi_m = 1$. Then

$$\mathrm{CVaR}_{\alpha,t,k} = \min_{z \in \mathbb{R}} \left\{ z + \frac{1}{1-\alpha} \sum_{m=1}^{M} \pi_m (\ell_m - z)^+ \right\}. \tag{11.7}$$

At an optimum, z is a VaR value [11, 12]. The advantage of Equation (11.7) for lattice models is that the lattice already provides a finite conditional loss table at every node. The formula handles unequal probabilities, discrete outcomes, and the splitting of a probability mass at the VaR point without

requiring a simulation model. It also links the risk-estimation step in this chapter to the CVaR-constrained optimization models used later in the book [8].

For hand calculation, Equation (11.7) is implemented by a sorted-tail calculation. Sort the conditional losses in increasing order, $\ell_{(1)} \le \ell_{(2)} \le \cdots \le \ell_{(M)}$, with associated probabilities $\pi_{(m)}$. Define the cumulative probabilities

$$\Pi_j = \sum_{m=1}^{j} \pi_{(m)}.$$

The VaR index j^* is the first index such that $\Pi_{j^*} \ge \alpha$, so

$$\mathrm{VaR}_{\alpha,t,k} = \ell_{(j^*)}.$$

CVaR then averages exactly the worst $1-\alpha$ probability mass. If the confidence level cuts through a probability mass at the VaR point, only the required fraction of that VaR atom is included. Let

$$q^+ = \sum_{m=j^*+1}^{M} \pi_{(m)}$$

be the probability strictly above the VaR point. The probability weight taken from the VaR atom is

$$q^0 = (1-\alpha) - q^+.$$

The sorted-tail formula is therefore

$$\mathrm{CVaR}_{\alpha,t,k} = \frac{q^0 \ell_{(j^*)} + \sum_{m=j^*+1}^{M} \pi_{(m)} \ell_{(m)}}{1-\alpha}. \tag{11.8}$$

Equations (11.7) and (11.8) are two ways to compute the same quantity. The Rockafellar–Uryasev representation is more compact and connects directly to optimization. The sorted-tail formula is more transparent for hand calculation, because it shows exactly which losses enter the tail average and how much of the VaR atom is included.

Copper application: baseline risk estimates

The copper data used in the examples are monthly nominal prices from the World Bank Commodity Price Data, commonly known as the Pink Sheet [15]. The sample runs from January 2000 to December 2024, and prices are measured in US dollars per metric ton. The diagnostics in Section 11.2

support a non-stationary random-walk baseline for the price level. The six-month lattice uses $u = 1.064$, $d = 0.940$, $\hat{p} = 0.549$, and the procurement loss in Equation (11.2).

Table 11.3: Copper data and baseline random-walk lattice inputs

Quantity	Value
Monthly price observations	300
Monthly log price changes	299
Current price, P_0	8916 dollars/metric ton
Mean monthly log price change	0.53%
Monthly log-price-change volatility, $\hat{\sigma}$	6.21%
Frequency of positive monthly log price changes, $\hat{p}$	54.85%
Up multiplier, $u = e^{\hat{\sigma}}$	1.064
Down multiplier, $d = e^{-\hat{\sigma}}$	0.940

© 2026 Fernando S. Oliveira

Worked Example 11.4: CVaR from the six-month copper lattice. At the initial node of the six-month random-walk copper lattice, the terminal loss table per metric ton is

Terminal loss	0	1179	2515	4027
Probability	0.562	0.277	0.134	0.027

The zero-loss probability combines all terminal nodes at which copper ends at or below the current price. The positive losses correspond to four, five, and six upward moves. This aggregation is the main advantage of the lattice: many paths are represented by a small conditional loss table, while the probabilities still reflect the number and likelihood of the paths reaching each terminal loss.

At the 95% confidence level, the cumulative probability reaches 0.839 after loss 1179 and 0.973 after loss 2515. Hence

$$\mathrm{VaR}_{0.95} = 2515.$$

The worst 5% probability mass contains all 2.7% probability at loss 4027 and only 2.3% probability from the VaR atom at loss 2515. Using the sorted-tail formula in Equation (11.8),

$$\mathrm{CVaR}_{0.95} = \frac{0.0228(2515) + 0.0272(4027)}{0.050} \approx 3338.$$

The same result is obtained from the Rockafellar–Uryasev representation in Equation (11.7). At $z = 2515$,

$$z + \frac{1}{1-0.95}\sum_m \pi_m(\ell_m - z)^+ = 2515 + \frac{1}{0.05}\left[0.0272(4027 - 2515)\right]$$
$$\approx 3338.$$

The minimising z is a VaR value. The sorted-tail calculation shows the tail composition; the Rockafellar–Uryasev formula gives the same CVaR through a single convex expression.

The same calculation produces a compact comparison with empirical six-month benchmarks. The benchmarks are included as context, not as substitutes for the lattice. The lattice is useful because it gives conditional future loss tables at each node, not only one empirical distribution estimated from past windows.

© 2026 Fernando S. Oliveira

Table 11.4: Six-month copper downside-risk estimates per metric ton

Estimate	Expected loss	VaR 95%	CVaR 95%
Random-walk lattice	774	2515	3338
Historical benchmark	497	2283	2712
Recent benchmark	657	2018	2662

Notes: The historical benchmark uses all six-month windows in 2000–2024. The recent benchmark uses the last 60 six-month windows.

For a planned purchase of 100 metric tons, the lattice CVaR exposure is

$$100 \times 3338 = 333800$$

dollars of additional procurement cost. This is not a forecast of the most likely cost. It is the average loss in the worst 5% of lattice-implied six-month outcomes.

Node-conditional risk maps

The same lattice can display the risk estimate at each node. Figure 11.4 shows the node-conditional expected terminal loss. Figure 11.5 shows the node-conditional 95% CVaR. The expected-loss lattice summarizes average exposure; the CVaR lattice emphasizes escalation states where the remaining tail loss is materially larger.

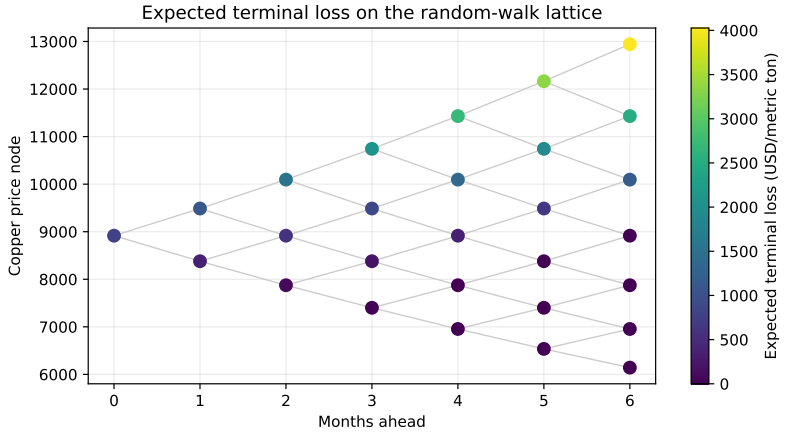

Figure 11.4: Node-conditional expected terminal loss on the six-month random-walk copper lattice.

© 2026 Fernando S. Oliveira

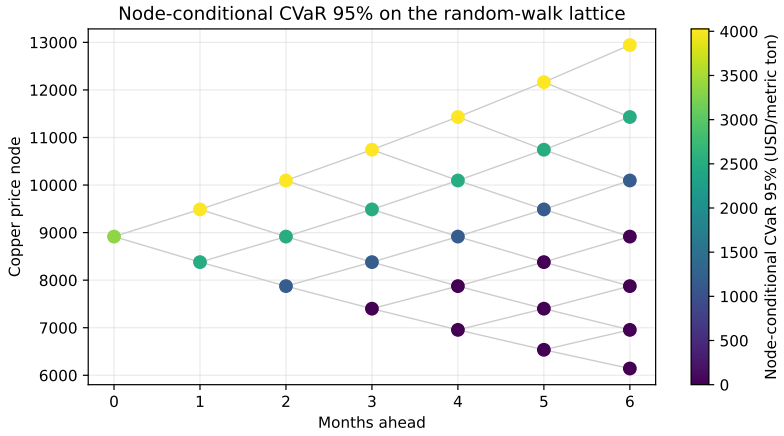

Figure 11.5: Node-conditional 95% CVaR on the six-month random-walk copper lattice.

In operational risk terms, the CVaR lattice is an escalation map. It identifies states where procurement risk has increased before a procurement, hedging, or inventory decision is optimized. It turns the lattice into a transparent monitoring device rather than a single static risk number.

11.6 Sensitivity Analysis and Model Transparency

The lattice is transparent because its assumptions can be stressed. Two sensitivities are most useful for the non-stationary random-walk case: the procurement horizon and the volatility calibration.

© 2026 Fernando S. Oliveira

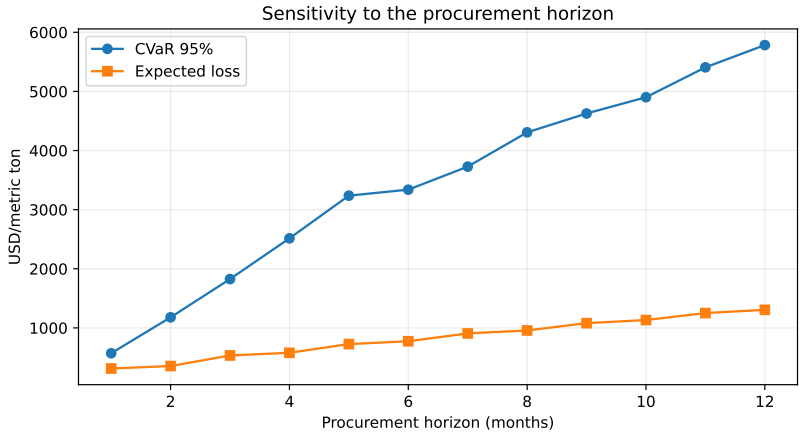

Figure 11.6: Sensitivity of copper downside risk to the procurement horizon.

Figure 11.6 shows that horizon matters. In the random-walk lattice, the 95% CVaR rises quickly as the horizon increases because shocks can compound. This is why procurement timing is a dynamic risk question rather than a static price question.

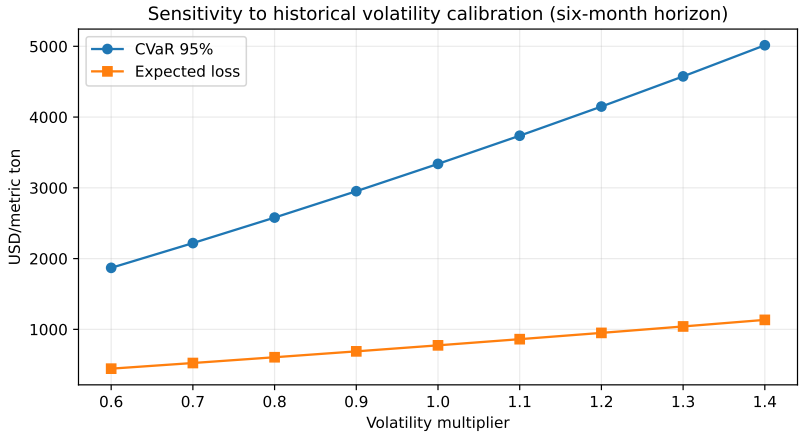

Figure 11.7: Sensitivity of six-month random-walk risk estimates to the volatility calibration.

Figure 11.7 uses the same six-month horizon as the baseline case and varies the historical volatility estimate by a multiplier. This is a model-risk check, not a forecast exercise. Volatility is estimated from a finite sample and

may change across regimes. Stressing it shows how much the CVaR estimate depends on the assumed scale of monthly price movements. Increasing the volatility multiplier from 0.8 to 1.2 raises the six-month random-walk CVaR from approximately 2579 to 4149 dollars per metric ton.

A final warning is essential. This chapter computes the CVaR of future losses conditional on each node. That is different from solving a fully dynamic CVaR optimization problem. In multi-stage decision problems, CVaR may be time inconsistent: an action that appears optimal from the initial date may not remain optimal after new information arrives. Recent work on CVaR in Markov decision processes emphasizes precisely this difficulty and explains why classical dynamic programming does not apply directly to CVaR objectives [16, 14, 9]. The chapter therefore remains an estimation chapter: it builds transparent conditional loss distributions before any optimisation layer is added.

11.7 Conclusions and Key Takeaways

This chapter converted a non-stationary commodity price series into a transparent dynamic risk map. The central step was the construction of conditional loss tables at the nodes of a price lattice. Once the price reaches a node, expected loss, VaR, and CVaR are computed from the losses still reachable from that node.

The copper example illustrates the logic. The log price level is persistent and the unit-root null is not rejected, while monthly log price changes are stable enough to calibrate the lattice. The resulting random-walk lattice preserves price-level persistence and translates it into procurement downside risk.

Key takeaways

- Test the price level before choosing the lattice. If the level is persistent, do not impose fast reversion to a historical mean.
- Use the lattice to construct conditional loss tables. These tables are the basis for expected loss, VaR, and CVaR at each node.
- Compute CVaR from the tail of the conditional loss table. The Rockafellar–Uryasev formula and the sorted-tail calculation give the same result; the latter makes atom splitting visible.
- Treat sensitivity analysis as part of the result. Horizon and volatility assumptions materially affect the copper CVaR estimate.

© 2026 Fernando S. Oliveira

Bibliography

[1] Peter J. Brockwell and Richard A. Davis. *Introduction to Time Series and Forecasting*. Springer Texts in Statistics. Springer, Cham, 3 edition, 2016.

[2] Gonzalo Cortazar, Ivo Kovacevic, and Eduardo S. Schwartz. Commodity and asset pricing models: An integration. NBER Working Paper 19167, National Bureau of Economic Research, 2013.

[3] John C. Cox, Stephen A. Ross, and Mark Rubinstein. Option pricing: A simplified approach. *Journal of Financial Economics*, 7(3):229–263, 1979.

[4] David A. Dickey and Wayne A. Fuller. Distribution of the estimators for autoregressive time series with a unit root. *Journal of the American Statistical Association*, 74(366):427–431, 1979.

[5] Paul Glasserman. *Monte Carlo Methods in Financial Engineering*. Springer, New York, 2004.

[6] James D. Hamilton. *Time Series Analysis*. Princeton University Press, Princeton, NJ, 1994.

[7] John C. Hull. *Options, Futures, and Other Derivatives*. Prentice Hall, Boston, 8 edition, 2012.

[8] Pavlo Krokhmal, Jonas Palmquist, and Stanislav Uryasev. Portfolio optimization with conditional value-at-risk objective and constraints. *The Journal of Risk*, 4(2):43–68, 2002.

[9] Georg Ch. Pflug and Alois Pichler. *Multistage Stochastic Optimization*. Springer, Cham, 2015.

[10] Craig Pirrong. *Commodity Price Dynamics: A Structural Approach*. University of Houston, Bauer College of Business, 2011. Manuscript dated October 11, 2015.

© 2026 Fernando S. Oliveira

[11] R. Tyrrell Rockafellar and Stanislav Uryasev. Optimization of conditional value-at-risk. *Journal of Risk*, 2(3):21–42, 2000.

[12] R. Tyrrell Rockafellar and Stanislav Uryasev. Conditional value-at-risk for general loss distributions. *Journal of Banking & Finance*, 26(7):1443–1471, 2002.

[13] Sergiy Sarykalin, Gian Paolo Serraino, and Stanislav Uryasev. Value-at-risk vs. conditional value-at-risk in risk management and optimization. *Tutorials in Operations Research*, pages 270–294, 2008.

[14] Alexander Shapiro. On a time consistency concept in risk averse multi-stage stochastic programming. *Operations Research Letters*, 37(3):143–147, 2009.

[15] World Bank. Commodity Price Data (The Pink Sheet): Monthly Prices. Excel workbook, CMO-Historical-Data-Monthly.xlsx, 2025. Monthly commodity prices in nominal U.S. dollars. Dataset used in Chapter 10 case study.

[16] Li Xia and Peter W. Glynn. Risk-sensitive markov decision processes with long-run cvar criterion. *arXiv preprint arXiv:2210.08740*, 2022.

© 2026 Fernando S. Oliveira