1 Loss_distribution_VaR_CVaR.py

Purpose

This script simulates portfolio credit losses to estimate two risk measures used heavily in banking regulation and trading risk limits:

- **Value at Risk (VaR)** — a percentile cut-off for losses at a chosen confidence level (e.g., 99%).
- **Conditional Value at Risk (CVaR)** — the *average* loss once you're already past the VaR threshold.

Learning focus

- VaR answers: "With 99% confidence, we expect losses not to exceed X."
- CVaR answers: "If we are unlucky enough to be in that worst 1% of scenarios, the *average* loss will be Y." The gap between CVaR and VaR tells you how "fat" the loss tail is.

Experiment

Change std_dev_loss to a higher value. Notice how both VaR and CVaR grow, but CVaR increases proportionally more. This is tail risk in action.

2 Survival_probability_curve.py

Purpose

Plots **survival probabilities** under constant hazard rates (λ). In credit risk, survival probability means: "What's the chance a borrower hasn't defaulted by time *t*?"

Learning focus

- Lower λ = safer credit = slower decay in survival probability.
- Higher λ = riskier credit = faster decay.
- These curves are used to price credit default swaps (CDS) and bonds, since expected cash flows are weighted by survival probability.

Experiment

Plot $\lambda = 0.01$ and $\lambda = 0.10$ on the same chart and note how quickly the risky borrower's survival curve hits 50%. That crossing time is a quick mental measure of credit quality.

3 Tranche_expected_loss.py

Purpose

Simulates a synthetic CDO with equity and mezzanine tranches using the **Gaussian copula model** to see how asset correlation affects expected loss distribution.

Learning focus

- Equity tranches (first-loss) absorb defaults early and are almost always hit — correlation has little effect when defaults are widespread.
- Mezzanine tranches are "safe" in low correlation environments, but when defaults cluster due to high correlation, losses jump sharply.
- Correlation risk means even if individual default probabilities are small, systemic events can wipe out supposedly safe tranches.

Experiment

Set correlation to 1.0. Notice how both tranches can be wiped out simultaneously — a pure systemic shock.

4 CDS_Spread_vs_Default_Probability.py

Purpose

Shows the **credit triangle**:

$$CDS\ spread = Default\ Probability * (1 - Recovery\ Rate)$$

This links credit market pricing to default likelihood and expected recovery.

Learning focus

- At fixed recovery, spreads rise linearly with default probability.
- Higher recovery rates (better collateral or stronger legal protections) reduce spreads for the same default risk.

Experiment

Double the recovery rate from 20% to 40% and watch the spread curve flatten — this is why senior secured loans trade tighter than unsecured bonds from the same issuer.

5 Credit_rating_transition_matrix_heatmap.py

Purpose

Creates a heatmap of a 1-year **credit rating transition matrix** — a table showing the probability of migrating from one rating to another in a year.

Learning focus

- Strong diagonal dominance = ratings tend to persist over a year.
- Downgrades are more common than upgrades.
- Default probability increases sharply as you move down the rating scale.

Experiment

Sum each row to confirm it's 100%. Then focus on the CCC row — notice how the default probability is vastly higher than for BBB, explaining yield spreads in speculative grade markets.

6 Expected_loss_vs_recovery_rate.py

Purpose

Shows how **Expected Credit Loss (ECL)** falls as recovery rates increase for a fixed loan notional and default probability.

Learning focus

- ECL falls linearly with recovery because loss given default = 1 – recovery.
- Recovery is a key driver in loan pricing and reserve setting.

Experiment

Keep default probability at 3% but vary recovery between 0% and 80%. This helps explain why secured loans (with higher recovery) require smaller credit provisions than unsecured ones.