L07: Risk Management & Regulation

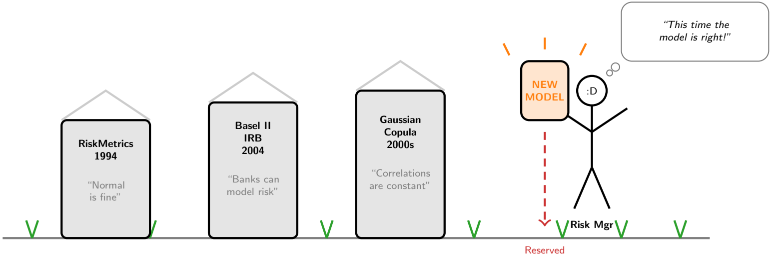
Extended Slides – BSc Digital Finance Course

Digital Finance

What Will You Be Able to Do After This Lecture?

1. Formally derive parametric VaR and Expected Shortfall from portfolio return distributions and prove ES subadditivity
2. Implement historical VaR, Monte Carlo VaR, and ES calculations in Python
3. Model credit portfolio losses using copula dependence and simulate loss distributions via Monte Carlo
4. Design reverse stress tests and quantify model risk using multiple model comparison
5. Calculate risk-weighted assets (RWA) under standardized and internal models approaches, and explain FRTB
6. Evaluate ML-based credit scoring using ROC/Gini metrics and implement operational risk quantification via Loss Distribution Approach

Six objectives: formal derivations (1, 4), Python implementations (2, 6), and applied evaluation (3, 5). This lecture combines rigorous math with working code and 11 data visualizations.



Every generation of risk managers believes their model is the one that finally got it right.

How Do You Model the Full Distribution of Losses in a Credit Portfolio?

Expected Loss (from main lecture):

$$\text{EL} = \sum_{i=1}^{n} \text{PD}_i \times \text{LGD}_i \times \text{EAD}_i$$

Unexpected Loss: $\text{UL} = \text{VaR}_\alpha(L) - \text{EL}$

Single-factor Vasicek model:

$$P_i(\text{default}) = \Phi\left(\frac{\Phi^{-1}(\text{PD}_i) - \sqrt{\rho}\,Z}{\sqrt{1-\rho}}\right), \quad Z \sim \mathcal{N}(0,1) \text{ (systematic factor)}$$

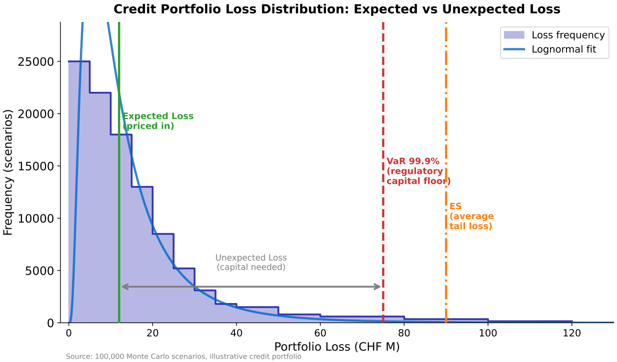
Portfolio loss: $L = \sum_{i=1}^{n} \text{LGD}_i \times \text{EAD}_i \times \mathbf{1}_{X_i \le \Phi^{-1}(\text{PD}_i)}$

Basel II regulatory formula:

$$K = \text{LGD} \times \left[\Phi\left(\frac{\Phi^{-1}(\text{PD}) + \sqrt{\rho}\,\Phi^{-1}(0.999)}{\sqrt{1-\rho}}\right) - \text{PD}\right]$$

The Vasicek model reduces a portfolio of 10,000 loans to a single systematic factor. The correlation parameter ρ is the most important – and hardest to estimate – input.

What Shape Does the Loss Distribution Actually Take – and Why Does It Matter?



- Reading the step histogram: each bar shows how many of 100,000 simulated scenarios fell into that loss bucket
- EL (expected loss) is the mean – priced into lending margins and is not the concern
- UL (unexpected loss) = VaR minus EL – this is the capital the bank must hold
- The tail is asymmetric: losses can far exceed the mean while gains are bounded
- Basel uses 99.9% confidence (1-in-1000 year event) to set the capital requirement

The step histogram reveals credit loss asymmetry: the right tail extends far beyond the mean. Economic capital = VaR minus EL – the gap between what you expect and what could happen.

Can You Simulate 100,000 Credit Portfolios in 15 Lines of Python?

```python
import numpy as np
from scipy.stats import norm

def mc_credit_loss(n_loans, pd, lgd, ead, rho, n_sims=100_000):
    """Single-factor Vasicek credit portfolio simulation."""
    Z = np.random.normal(size=n_sims)       # systematic factor
    eps = np.random.normal(size=(n_sims, n_loans)) # idiosyncratic
    X = np.sqrt(rho)*Z[:,None] + np.sqrt(1-rho)*eps
    defaults = (X < norm.ppf(pd)).astype(float)
    losses = defaults * lgd * ead           # per-loan loss
    portfolio_loss = losses.sum(axis=1)     # total loss
    return portfolio_loss

losses = mc_credit_loss(n_loans=1000, pd=0.02, lgd=0.4,
                      ead=1.0, rho=0.15)
el = losses.mean()
var99 = np.percentile(losses, 99.9)
es99  = losses[losses >= var99].mean()
print(f"EL={el:.2f} VaR(99.9%)={var99:.2f} ES={es99:.2f}")
```

Fifteen lines of Python simulate 100,000 portfolios of 1,000 loans. The correlation parameter ρ=0.15 is what Basel uses for corporate exposures – change it and watch the tail explode.

Why Do Correlations Lie About Tail Dependence – and What Replaces Them?

Sklar's theorem:

$$F(x_1, \dots, x_n) = C(F_1(x_1), \dots, F_n(x_n))$$

Gaussian copula: $C_G(u_1, u_2) = \Phi_2(\Phi^{-1}(u_1), \Phi^{-1}(u_2); \rho)$

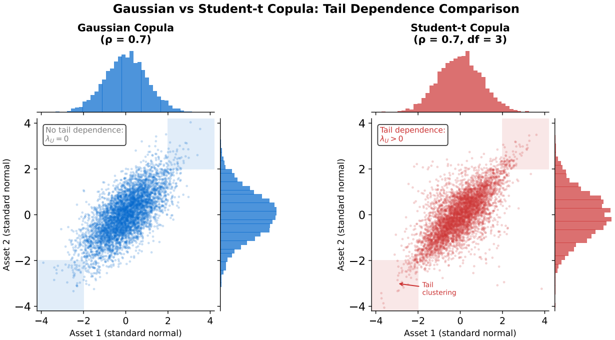
Upper tail dependence: $\lambda_U = \lim_{u \to 1} P[U_2 > u \mid U_1 > u]$

- Gaussian copula: $\lambda_U = 0$ (zero tail dependence – this is why it failed in 2008)
- Student-t copula: $\lambda_U = 2\, t_{\nu+1}\left(-\sqrt{\frac{(\nu+1)(1-\rho)}{1+\rho}}\right) > 0$

The "formula that killed Wall Street": pricing CDOs with Gaussian copula assumed defaults could not cluster. When they did, the model collapsed.

The Gaussian copula has zero tail dependence – it says simultaneous extreme events are essentially impossible. The 2008 crisis proved otherwise.

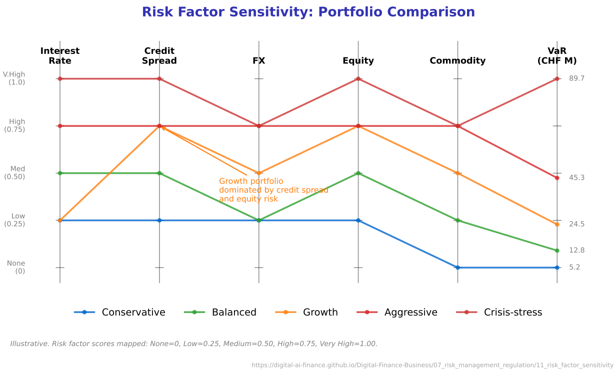
How Does the Choice of Copula Change What You See in the Tails?



- Two panels: Gaussian copula (left) vs Student-t copula (right), both with ρ=0.7
- Gaussian panel: points scatter uniformly – no clustering in the corners (zero tail dependence)
- Student-t panel: points cluster in bottom-left and top-right corners – joint extreme events are far more likely
- The marginal histograms look similar, but the joint tails are radically different
- This visual explains why Gaussian copula CDO pricing catastrophically underestimated correlated defaults

Same correlation, same marginals, different tails. The Gaussian copula says joint extremes are nearly impossible; the t-copula says they cluster. The 2008 crisis proved the t-copula right.

Which Risk Factors Drive the Most Variation in Portfolio Loss?



- Each vertical axis represents one risk factor, normalized to 0–1; each colored line is one portfolio
- Conservative portfolio (blue): flat low profile across all factors – minimal exposure, minimal VaR
- Growth portfolio (orange): spikes on credit spread and equity axes – reveals concentration risk
- Crisis-stress portfolio (red): peaks on every axis – systemic exposure that amplifies correlated shocks
- The steepness between axes shows which factor transitions drive the most VaR

Parallel coordinates reveal risk factor profiles at a glance: the Growth portfolio's spike on credit spread and equity shows where concentrated exposure hides.

How Do You Derive VaR from a Return Distribution – and What Happens When It Is Not Normal?

Portfolio return: $R_p = \mathbf{w}'\boldsymbol{\mu}, \quad \sigma_p = \sqrt{\mathbf{w}'\Sigma\mathbf{w}}$

Parametric VaR (Gaussian):

$$\text{VaR}_\alpha = -(\mu_p + z_\alpha \sigma_p) \times V_0$$

At 99%: $z_{0.01} = -2.326$, so $\text{VaR}_{99\%} = (2.326\sigma_p - \mu_p) \times V_0$

Cornish–Fisher expansion for non-normality:

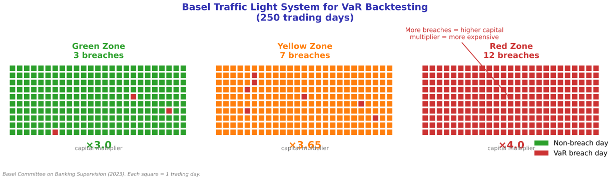
$$z_{CF} = z_\alpha + \frac{1}{6}(z_\alpha^2 - 1)S + \frac{1}{24}(z_\alpha^3 - 3z_\alpha)K - \frac{1}{36}(2z_\alpha^3 - 5z_\alpha)S^2$$

where S = skewness, K = excess kurtosis.

Example: $S = -0.5$, $K = 3$ (fat tails) inflates VaR by $\sim$15% vs Gaussian.

The Cornish–Fisher adjustment inflates VaR for fat-tailed distributions. Without it, Gaussian VaR underestimates the 99% loss by 10–20%.

How Many VaR Breaches Are Too Many – and What Happens When You Cross the Line?



https://digital-ai-finance.github.io/Digital-Finance-Business/07_risk_management_regulation/12_var_backtest_traffic_light

- Each grid represents 250 trading days; red squares mark VaR breach days
- Green zone (0–4 breaches): model accepted, capital multiplier stays at 3.0x
- Yellow zone (5–9 breaches): regulator investigates, multiplier rises to 3.4x–3.85x
- Red zone (10+ breaches): model rejected, multiplier jumps to 4.0x and bank must switch to standardized approach
- The jump from 4 to 5 breaches crosses a regulatory cliff edge

The Basel traffic light system is binary at the boundaries: 4 breaches is green, 5 is yellow, 10 is red. One extra breach can increase capital requirements by 13%.

Can You Calculate VaR and ES Three Different Ways in Python?

```python
import numpy as np
from scipy.stats import norm, t as student_t

def risk_metrics(returns, alpha=0.05):
    """Three VaR methods + ES."""
    n = len(returns)
    # 1. Historical
    var_hist = -np.percentile(returns, alpha * 100)
    es_hist = -returns[returns <= -var_hist].mean()
    # 2. Parametric (Gaussian)
    mu, sig = returns.mean(), returns.std()
    var_param = -(mu + norm.ppf(alpha) * sig)
    es_param = -(mu - sig * norm.pdf(norm.ppf(alpha)) / alpha)
    # 3. Student-t (fat tails)
    df = 5 # degrees of freedom
    var_t = -(mu + student_t.ppf(alpha, df) * sig)
    es_t  = -(mu - sig*(student_t.pdf(student_t.ppf(alpha,df),df)/alpha)
             * (df + student_t.ppf(alpha,df)**2) / (df - 1))
    return {'hist': (var_hist, es_hist), 'gauss': (var_param, es_param),
            't5': (var_t, es_t)}

r = np.random.standard_t(df=5, size=1000) * 0.015
for name, (v, e) in risk_metrics(r).items():
    print(f"{name:6s}: VaR={v:.4f} ES={e:.4f} ES/VaR={e/v:.2f}")
```

Three methods, one insight: Student-t VaR exceeds Gaussian VaR by 20–40% and ES/VaR ratio increases from 1.2 (Gaussian) to 1.5 (t with df=5). The fatter the tail, the bigger the gap.

Why Is VaR Not a Coherent Risk Measure – and Why Does That Matter?

Artzner et al. (1999) four axioms for a coherent risk measure ϱ:

1. **Translation invariance:** $\varrho(X + c) = \varrho(X) - c$
2. **Subadditivity:** $\varrho(X + Y) \leq \varrho(X) + \varrho(Y)$ – diversification cannot increase risk
3. **Positive homogeneity:** $\varrho(\lambda X) = \lambda\,\varrho(X)$ for $\lambda > 0$
4. **Monotonicity:** $X \leq Y \implies \varrho(X) \geq \varrho(Y)$

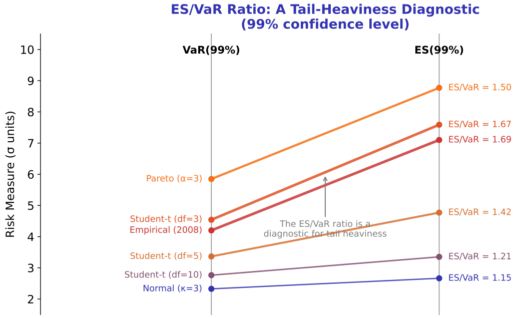
VaR violates subadditivity: counterexample with two independent binary losses shows $\mathrm{VaR}(X + Y) > \mathrm{VaR}(X) + \mathrm{VaR}(Y)$.

ES satisfies all four axioms: $\mathrm{ES}_\alpha(X + Y) \leq \mathrm{ES}_\alpha(X) + \mathrm{ES}_\alpha(Y)$

Practical consequence: a bank that merges two trading desks can see VaR increase even though actual risk decreased.

VaR violates subadditivity: merging two portfolios can increase VaR even when diversification reduces actual risk. This mathematical flaw is why Basel III replaced VaR with ES.

How Much More Does ES Reveal Than VaR – and When Does the Gap Explode?



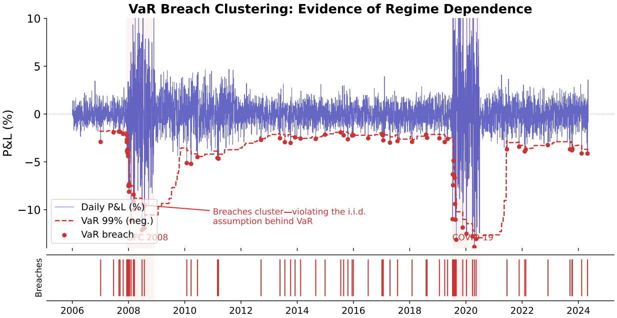
Illustrative σ-multiples at 99% VaR/ES. Thicker lines indicate larger ES/VaR ratios (heavier tails).

https://digital-ai-finance.github.io/Digital-Finance-Business/07_risk_management_regulation/13_es_var_ratio

- Each line connects a distribution's VaR (left) to its ES (right); steeper slope = fatter tails
- Normal distribution: ES/VaR = 1.15 – the tail is thin, ES barely exceeds VaR
- Student-t (df=5): ES/VaR = 1.42 – ES reveals 42% more risk than VaR hides
- Empirical 2008: ES/VaR = 1.69 – real crisis tails are heavier than any standard parametric model
- Diagnostic: if ES/VaR exceeds 1.4, your distribution has meaningful fat tails

The slope chart is a tail-heaviness diagnostic: steeper slope = fatter tails = more risk hidden beyond VaR. The 2008 empirical ratio of 1.69 means ES captured 69% more tail risk than VaR.

When Do VaR Breaches Cluster – and What Does Clustering Tell You?



- Top panel: daily P&L (purple line) with VaR threshold (red dashed); bottom rug marks breach dates
- Calm periods (2012–2019): breaches are rare and randomly scattered – the i.i.d. assumption holds
- Crisis periods (2008, 2020): breaches cluster in dense bursts – violating independence
- VaR threshold lags: it adapts slowly while losses spike instantly, creating breach cascades
- Breach clustering is the fingerprint of regime change

Breaches should be uniformly scattered if the model is correct. Clustering proves they are not – and the clusters coincide with exactly the periods where accurate risk measurement matters most.

What Scenario Would Actually Kill the Bank – and How Do You Find It?

Forward stress test: fix scenario $\mathbf{s}$, compute loss $L(\mathbf{s})$.

Reverse stress test: fix loss threshold L^* (e.g., bank failure), solve for scenario:

$$\mathbf{s}^* = \arg\min_{\mathbf{s}} \|\mathbf{s}\| \quad \text{s.t.} \quad L(\mathbf{s}) \geq L^*$$

This is a constrained optimization: find the "nearest" scenario to today that causes failure.

Mahalanobis distance:

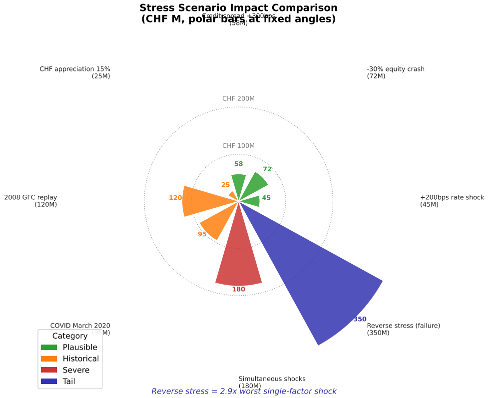
$$d(\mathbf{s}) = \sqrt{(\mathbf{s} - \boldsymbol{\mu})' \Sigma^{-1} (\mathbf{s} - \boldsymbol{\mu})}$$

The reverse stress test finds the most plausible catastrophic scenario.

Regulatory requirement: FINMA, EBA, PRA all require reverse stress testing.

Reverse stress testing inverts the question: instead of "how bad could it get?" it asks "what scenario would kill us?" – and then works backward to find the shortest path to failure.

How Do Different Stress Scenarios Compare in Severity and Plausibility?



- Each spoke represents one stress scenario; bar length = P&L impact magnitude
- Inner ring (short bars): plausible single-factor shocks with moderate impact
- Middle ring: historical replays (GFC, COVID) with severe but precedented impact
- Outer ring: simultaneous shocks and reverse stress scenarios approach bank failure thresholds
- The reverse stress scenario is 2–3x worse than any single historical crisis replay

The polar bar reveals a geometry of risk: single-factor shocks cluster in the inner ring, but combined shocks and reverse stress scenarios push into the outer ring where capital buffers break.

Can You Build a Multi-Factor Stress Test Engine in Python?

```python
import numpy as np

def stress_test(portfolio, scenarios, correlations):
    """Multi-factor stress test with correlated shocks."""
    n_factors = len(scenarios[0])
    L = np.linalg.cholesky(correlations)
    results = {}
    for name, shocks in scenarios.items():
        corr_shocks = L @ np.array(shocks)
        pnl = sum(w * s for w, s in
                  zip(portfolio['weights'], corr_shocks))
        results[name] = pnl * portfolio['notional']
    return results

portfolio = {'weights': [0.3, 0.25, 0.2, 0.15, 0.1],
             'notional': 1e9} # CHF 1B
corr = np.array([[1,.6,.3,-.2,.1],[.6,1,.4,-.1,.2],
    [.3,.4,1,.1,.3],[-.2,-.1,.1,1,-.3],[.1,.2,.3,-.3,1]])
scenarios = {'GFC': [-0.05,0.03,-0.04,-0.08,-0.02],
             'COVID': [-0.03,0.02,-0.06,-0.12,-0.01],
             'Rate+Equity': [0.04,-0.01,0.03,-0.06,0.0]}
for name, pnl in stress_test(portfolio, scenarios, corr).items():
    print(f"{name:15s}: CHF {pnl/1e6:+.1f}M")
```

The Cholesky decomposition ensures stress scenarios respect factor correlations. Without it, you assume rate shocks and equity crashes are independent – they are not.

How Do You Measure the Risk of the Risk Model Itself?

Model risk: $\text{MR} = |L_{\text{actual}} - L_{\text{model}}|$

Danielsson et al. (2016): "Models measure risk badly because risk is hard to measure, not because models are badly made."

Model uncertainty quantification: run K models, compute $\text{VaR}^{(k)}$ for each.

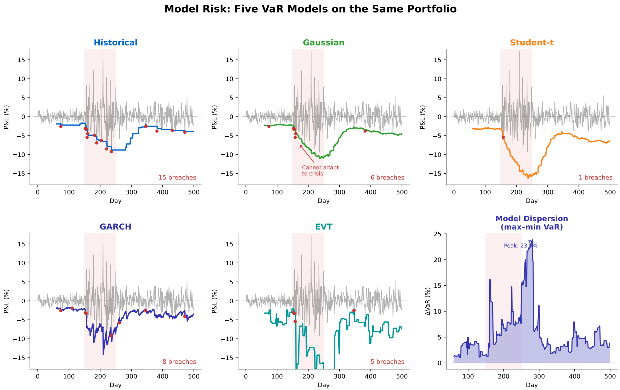
- **Model dispersion:** $\text{MD} = \max_k \text{VaR}^{(k)} - \min_k \text{VaR}^{(k)}$
- **Model confidence interval:** $[\text{VaR}_{\min}, \text{VaR}_{\max}]$ across models

P&L attribution: $\text{PnL}_{\text{explained}} = \sum_i \Delta_i \cdot \Delta f_i$ (greeks decomposition)

Unexplained P&L = actual − explained (should be small; large = model risk).

If five models give five different VaR numbers, the dispersion between them IS the model risk. The honest answer is not a single number – it is a range.

How Much Do Different Risk Models Disagree – and What Does the Disagreement Mean?



- Five panels show five VaR models applied to the same portfolio – each panel plots VaR forecast vs actual P&L
- Calm periods: all five models give similar VaR estimates (dispersion <0.5%)
- Crisis peak: dispersion explodes to 4.7% (Gaussian VaR = 1.5% vs GARCH VaR = 6.2%)
- Gaussian VaR is flat through the crisis; GARCH spikes ahead of the worst losses
- The dispersion between models IS the model risk

Model risk is measurable: the max-min VaR dispersion across five models jumps from 0.5% in calm markets to 4.7% in crises. That 4.7% gap is the honest answer about what we do not know.

How Do You Calculate Risk-Weighted Assets – and Why Does the Method Matter?

Standardized approach: $\text{RWA} = \sum_i w_i \times \text{EAD}_i$

Risk weights: sovereign (0–150%), bank (20–150%), corporate (20–150%), retail (75%), mortgage (35%).

IRB approach: $\text{RWA} = 12.5 \times K \times \text{EAD}$ where K is from Vasicek formula.

FRTB market risk (Internal Models Approach):

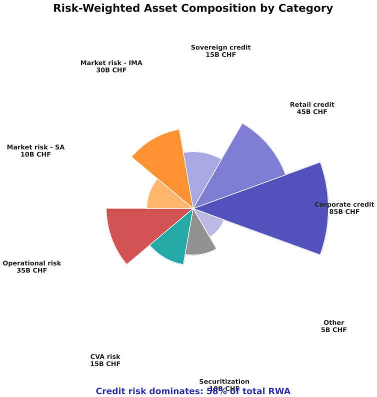
$$\text{IMA} = \max\left(\text{ES}_{t-1},\ m_c \cdot \frac{1}{60}\sum_{i=1}^{60} \text{ES}_{t-i}\right) + \text{DRC} + \text{SES}$$

where $m_c \geq 1.5$ is the multiplier (increases with backtesting failures), DRC = default risk charge, SES = stress ES.

Output floor: $\text{RWA}_{\text{final}} = \max(\text{RWA}_{\text{IMA}},\ 72.5\% \times \text{RWA}_{\text{SA}})$

The FRTB output floor at 72.5% ensures internal models cannot produce RWA less than 72.5% of the standardized approach – closing the gap that allowed banks to game capital requirements.

How Is a Typical Bank's Risk Capital Allocated Across Risk Types?



- Each wedge represents a risk category; wedge radius shows RWA magnitude in CHF billions
- Credit risk dominates: corporate (34%) + retail (18%) + sovereign (6%) = 58% of total RWA
- Market risk (IMA + SA = 16%) is smaller but grows fastest under FRTB (+40–90% increase)
- Operational risk (14%) is hardest to model – it includes cyber risk, fraud, and process failures
- The coxcomb reveals that credit risk capital is 3.6x market risk capital

Credit risk is the silent majority: 58% of RWA, yet FRTB focuses on market risk. The coxcomb reveals that capital allocation does not match regulatory attention.

Can You Build a Basel III Capital Adequacy Calculator in Python?

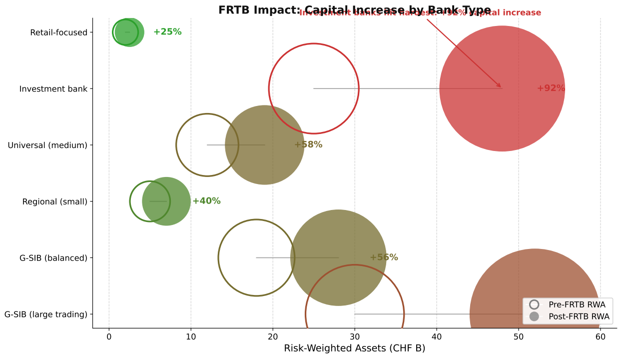
```python
import numpy as np

def capital_adequacy(exposures, risk_weights, capital):
    """Basel III capital adequacy check."""
    rwa = sum(e * w for e, w in zip(exposures, risk_weights))
    cet1_ratio = capital['cet1'] / rwa
    t1_ratio  = (capital['cet1'] + capital['at1']) / rwa
    total_ratio = sum(capital.values()) / rwa
    reqs = {'CET1': (cet1_ratio, 0.045), 'T1': (t1_ratio, 0.06),
            'Total': (total_ratio, 0.08)}
    buffer = capital['cet1'] / rwa - 0.07 # incl. conservation
    return rwa, reqs, buffer

exposures = [500, 300, 200, 100, 50] # CHF M
rw = [0.0, 0.20, 1.00, 0.75, 0.35] # sovereign,bank,corp,retail,mort
capital = {'cet1': 45, 'at1': 10, 't2': 15} # CHF M
rwa, ratios, buf = capital_adequacy(exposures, rw, capital)
print(f"RWA: CHF {rwa:.0f}M")
for name, (actual, req) in ratios.items():
    status = 'PASS' if actual >= req else 'FAIL'
    print(f"{name}: {actual:.1%} vs {req:.1%} [{status}]")
print(f"Conservation buffer: {buf:.1%}")
```

Five lines of arithmetic determine whether a bank meets Basel III requirements. The entire regulatory framework reduces to: does your capital ratio exceed the minimum?

How Much More Capital Will FRTB Require – and Which Banks Are Hit Hardest?



- Each paired circle shows pre-FRTB (hollow) vs post-FRTB (filled) market risk RWA; circle area = RWA size
- Investment banks hit hardest: +92% capital increase because trading books dominate their balance sheets
- G-SIBs with large trading operations: +73% – ES requirement and desk-level testing add significant capital
- Retail-focused banks: only +25% – small trading books mean FRTB barely touches them
- FRTB penalizes complexity – the more exotic the trading book, the larger the capital jump

FRTB capital increases range from +25% (retail banks) to +92% (investment banks). The regulation penalizes exactly what caused 2008: large, complex, model-dependent trading books.

How Do You Prove That a Machine Learning Credit Model Is Better Than a Human?

ROC curve: True Positive Rate vs False Positive Rate at each threshold.

$$\text{TPR}(t) = P(\hat{Y} \geq t \mid Y = 1), \quad \text{FPR}(t) = P(\hat{Y} \geq t \mid Y = 0)$$

AUC (Area Under Curve): $\text{AUC} = \int_0^1 \text{TPR}(\text{FPR}^{-1}(x))\, dx$

- $\text{AUC} = 0.5$: random model. $\text{AUC} = 1.0$: perfect model. Good credit model: $\text{AUC} > 0.75$

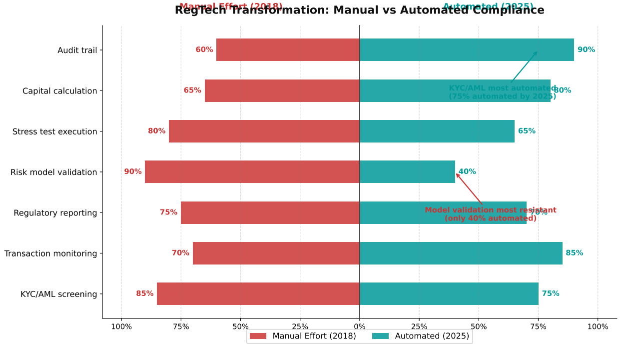
Gini coefficient: $\text{Gini} = 2 \times \text{AUC} - 1$
Accuracy Ratio: $\text{AR} = \frac{\text{AUC}_{\text{model}} - 0.5}{0.5}$

Information Value: $\text{IV} = \sum_i (D_i - ND_i) \times \ln\left(\frac{D_i}{ND_i}\right)$

ML vs traditional: XGBoost AUC $\sim$0.82 vs logistic regression AUC $\sim$0.75 on same data.

AUC measures discrimination: can the model distinguish defaulters from non-defaulters? A 7-point AUC improvement (0.75 to 0.82) can reduce credit losses by 15–20%.

How Rapidly Is RegTech Replacing Manual Compliance – and What Functions Lead?



- Left bars (red) show manual effort in 2018; right bars (teal) show automated effort in 2025
- KYC/AML and audit trails led automation: manual effort dropped from 85%/60% to 25%/10%
- Risk model validation resists: still 60% manual in 2025 – validating models requires human judgment
- Transaction monitoring shifted fastest (70% to 15%) because ML anomaly detection outperforms human reviewers
- RegTech automates the routine but cannot replace judgment-intensive tasks

RegTech automated the easy tasks first: KYC screening dropped from 85% to 25% manual. But risk model validation stays 60% manual – because validating assumptions requires the judgment machines lack.

Can You Quantify Operational Risk Using the Loss Distribution Approach?

```python
import numpy as np
from scipy.stats import poisson, lognorm

def lda_simulation(freq_lambda, sev_mu, sev_sigma, n_sims=100_000):
    """Loss Distribution Approach for operational risk."""
    total_losses = np.zeros(n_sims)
    for i in range(n_sims):
        n_events = poisson.rvs(freq_lambda)
        if n_events > 0:
            severities = lognorm.rvs(s=sev_sigma, scale=np.exp(sev_mu),
                                     size=n_events)
            total_losses[i] = severities.sum()
    return total_losses

# Simulate: avg 5 events/year, severity ~CHF 2M median
losses = lda_simulation(freq_lambda=5, sev_mu=14.5, sev_sigma=1.5)
el = losses.mean()
var99 = np.percentile(losses, 99.9)
es99  = losses[losses >= var99].mean()
print(f"EL: CHF {el/1e6:.1f}M")
print(f"OpRisk VaR(99.9%): CHF {var99/1e6:.1f}M")
print(f"OpRisk ES(99.9%): CHF {es99/1e6:.1f}M")
```

The Loss Distribution Approach compounds frequency (how often?) with severity (how bad?) to get the operational risk loss distribution. The tail is driven by severity, not frequency.

What Have We Learned – and What Remains Unsolved?

1. **Quantitative Risk Modeling:** Credit losses follow a skewed distribution driven by default correlations. The Vasicek single-factor model underpins Basel IRB, but its Gaussian copula assumption underestimates tail dependence.
2. **VaR & ES Framework:** VaR answers "how bad?" but not "how much worse?" ES is coherent (subadditive) and captures tail severity. Basel III's shift from VaR to ES corrects a mathematical flaw.
3. **Stress Testing & Model Validation:** Forward stress tests check plausible scenarios; reverse stress tests find the shortest path to failure. Model risk – the disagreement between models – is itself a measurable quantity.
4. **Capital Adequacy & FRTB:** FRTB replaces VaR with ES, adds desk-level P&L attribution testing, and imposes an output floor. Capital for trading books will increase 40–90%.
5. **Digital Risk & RegTech:** ML credit models improve discrimination (AUC 0.75 $\rightarrow$ 0.82) but sacrifice explainability. RegTech automates compliance but creates dependency on automated systems that themselves need validation.

Unsolved: Explainable AI for credit decisions, quantum computing impact on risk simulation, climate risk integration into stress testing, regulatory harmonization across jurisdictions.

The model paradox was never solved – it was distributed across more models, more data, and more compute. The fundamental challenge remains: risk models work until they do not.

Key Takeaways

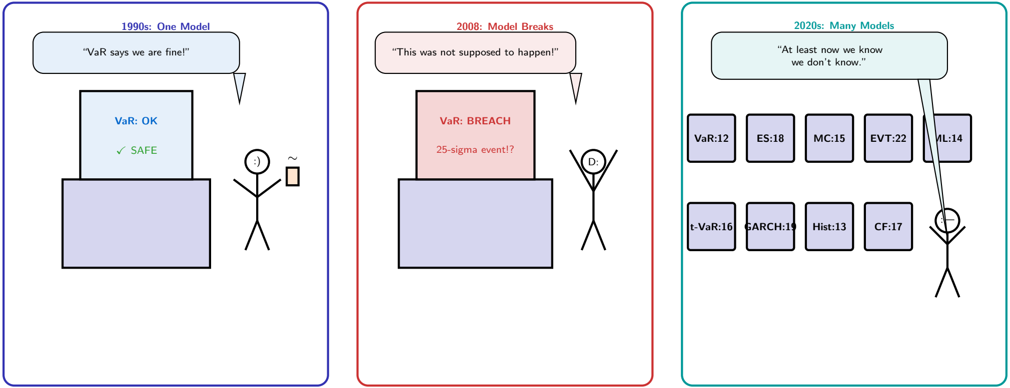
1. **Credit portfolio losses are driven by correlations, not individual defaults.** The Vasicek model reduces 10,000 loans to one systematic factor. The correlation parameter spikes in crises.
2. **VaR is not coherent; ES is.** VaR violates subadditivity. ES satisfies all four Artzner axioms and captures tail severity. Basel III's shift from VaR to ES corrects this mathematical flaw.
3. **Reverse stress testing inverts the question.** Instead of "how bad could scenario X be?" it asks "what scenario kills the bank?" and works backward. The result is the most plausible catastrophic scenario.
4. **FRTB increases trading book capital by 40–90%.** ES replaces VaR, desk-level testing adds granularity, and the 72.5% output floor prevents internal models from gaming capital requirements.
5. **ML credit models outperform but sacrifice explainability.** A 7-point AUC improvement reduces credit losses by 15–20%. But regulators demand banks explain why a loan was denied.
6. **Model risk is the risk of the risk model.** When five models give five different VaR numbers, the dispersion between them is the model risk. The honest answer is a range, not a single number.

Six takeaways, one framework: risk measurement is a precision-robustness tradeoff. The models that appear most precise are often the most fragile.

References and Further Reading

1. Artzner, P., Delbaen, F., Eber, J.-M., & Heath, D. (1999). "Coherent Measures of Risk." *Mathematical Finance*, 9(3), 203–228. *The foundational paper defining subadditivity and coherent risk measures.*
2. McNeil, A.J., Frey, R., & Embrechts, P. (2015). *Quantitative Risk Management: Concepts, Techniques and Tools.* Princeton University Press. *The standard graduate textbook covering VaR, ES, copulas, and EVT.*
3. Basel Committee on Banking Supervision (2019). "Minimum capital requirements for market risk." *BCBS d457. The FRTB framework replacing VaR with ES for market risk capital.*
4. Danielsson, J. (2011). *Financial Risk Forecasting.* Wiley. *Practical guide to VaR implementation with emphasis on model risk and backtesting.*
5. Li, D.X. (2000). "On Default Correlation: A Copula Function Approach." *Journal of Fixed Income*, 9(4), 43–54. *The Gaussian copula paper that became the "formula that killed Wall Street."*
6. Vasicek, O. (2002). "The Distribution of Loan Portfolio Value." *Risk*, 15(12), 160–162. *The single-factor model underlying Basel II/III IRB capital requirements.*

Start with Artzner (1999) for the theory, McNeil et al. (2015) for the textbook treatment, and BCBS d457 (2019) for the regulatory framework.



One model gave false confidence. Many models give honest uncertainty. Progress is not better answers – it is better questions.