L07: VaR & Expected Shortfall – Advanced Quantitative Methods

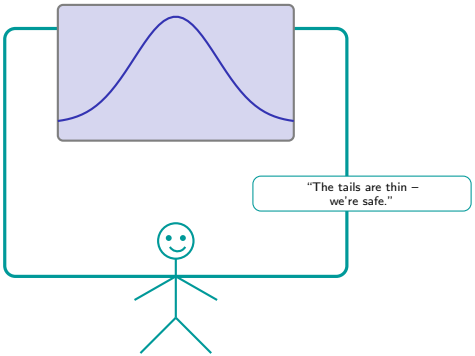
Extended Slides – BSc Digital Finance Course

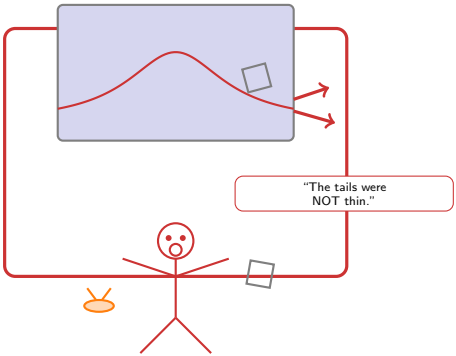
Digital Finance

What Will You Be Able to Do After This Lecture?

1. Fit Generalized Pareto distributions to portfolio tail losses and compute EVT-based VaR/ES
2. Implement Filtered Historical Simulation with GARCH volatility scaling
3. Calculate delta-gamma VaR for option portfolios using full revaluation and Taylor approximation
4. Decompose portfolio VaR into marginal, component, and incremental contributions using Euler allocation
5. Conduct formal VaR backtests using Kupiec, Christoffersen, and Berkowitz tests
6. Apply Acerbi-Szekely tests to backtest Expected Shortfall and explain why ES requires joint elicitability with VaR

Six objectives spanning EVT (1), simulation (2), non-linear risk (3), attribution (4), VaR backtesting (5), and ES backtesting (6). Theory with code and 12 visualizations.





Every tail looks thin until it bites you.

Why Does Extreme Value Theory Let You Model Tails Without Knowing the Full Distribution?

Fisher-Tippett-Gnedenko Theorem: Block maxima $M_n = \max(X_1, \ldots, X_n)$ converge in distribution to the Generalized Extreme Value (GEV) family:

$$H_\xi(x) = \exp\left\{-\left[1+\xi\left(\frac{x-\mu}{\sigma}\right)\right]^{-1/\xi}\right\}, \quad 1+\xi\left(\frac{x-\mu}{\sigma}\right) > 0$$

Three domains of attraction determined by the shape parameter ξ (tail index):

- $\xi > 0$: **Fréchet** (fat tails) – financial returns live here. Tails decay as a power law $\sim x^{-1/\xi}$
- $\xi = 0$: **Gumbel** (thin tails) – exponential-type decay (e.g., Normal distribution)
- $\xi < 0$: **Weibull** (bounded tails) – not relevant for financial losses

EVT-based VaR (where α is the tail probability, e.g., $\alpha = 0.01$ for 99% VaR):

$$\text{VaR}_\alpha = \mu + \frac{\sigma}{\xi}\left[(-\ln(1-\alpha))^{-\xi} - 1\right]$$

The Fisher-Tippett theorem is remarkable: regardless of the parent distribution, block maxima converge to one of three families. For financial returns, $\xi > 0$ (Fréchet) – tails are always fat. Here α is the tail probability ($\alpha = 0.01$ for 99% VaR).

How Do You Extract More Information from Tail Observations Than Block Maxima Allow?

Peaks-over-Threshold (POT): condition on losses exceeding a high threshold u.
Pickands-Balkema-de Haan Theorem: exceedances $Y = X - u \mid X > u$ converge to GPD:

$$G_{\xi,\beta}(y) = 1 - \left(1 + \frac{\xi y}{\beta}\right)^{-1/\xi}, \quad y > 0, \quad 1 + \frac{\xi y}{\beta} > 0$$

GPD-based risk measures (α = tail probability, N_u = number of exceedances, n = total obs):

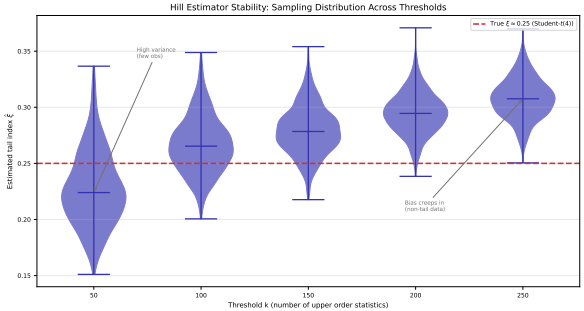
$$\mathrm{VaR}_\alpha = u + \frac{\beta}{\xi}\left[\left(\frac{n}{N_u}\alpha\right)^{-\xi} - 1\right]$$

$$\mathrm{ES}_\alpha = \frac{\mathrm{VaR}_\alpha}{1-\xi} + \frac{\beta - \xi\, u}{1-\xi}, \qquad \text{valid only if } \xi < 1$$

- Constraint: $\xi < 1$ for ES to exist (finite mean of exceedances)
- Threshold selection: bias-variance tradeoff – too low $\Rightarrow$ bias, too high $\Rightarrow$ variance

POT uses every extreme observation, not just block maxima – extracting far more information from limited tail data. The tradeoff: the threshold u must be high enough for the GPD approximation to hold, but low enough to have sufficient data.

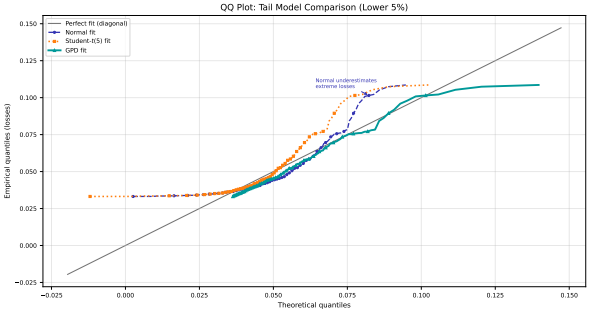
How Stable Is the Tail Index Estimate – and When Should You Stop Trusting It?



- Each violin shows sampling variability of the Hill estimator at one threshold level
- Low k (few extremes): wide violins – high variance, unreliable
- High k (too many): violins shift – bias from non-tail data
- Optimal k: narrow violin centered on true ξ – the bias-variance sweet spot
- For S&P 500: $\hat{\xi} \approx 0.25$–0.35 (4th moment barely exists)

The Hill plot is the EVT practitioner's diagnostic: a stable plateau in $\hat{\xi}$ across thresholds means the tail model is trustworthy. No plateau means the tail index is unknowable at this sample size.

Can You Build a Risk Model That Is Nonparametric in the Body but Parametric in the Tails?



- QQ plot compares model quantiles (x-axis) vs empirical quantiles (y-axis)
- Points on the diagonal = perfect fit
- Normal fit (purple dashed): curves away – underestimates tail losses
- Student-t fit (orange): closer but still misses the deepest extremes
- GPD tail fit (teal solid): hugs the diagonal even at the 0.1% quantile
- Semi-parametric: use empirical CDF for the body, GPD for the tail

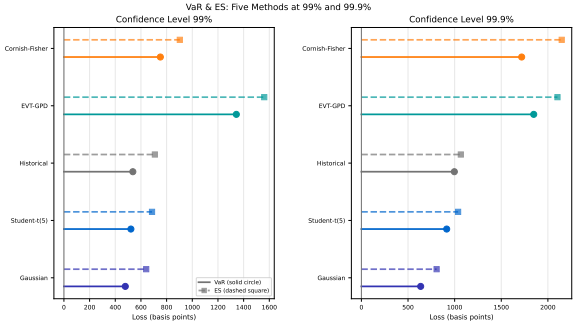
The QQ plot is the ultimate model diagnostic: points that peel away from the diagonal reveal exactly where the model fails. The GPD tail fit stays on the diagonal where it matters – in the extremes.

Can You Fit a Generalized Pareto Distribution to Tail Losses in 20 Lines?

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

def evt_var_es(returns, alpha=0.01, quantile_threshold=0.95):
    """EVT-based VaR and ES using Peaks-over-Threshold.
    alpha = tail probability (0.01 for 99% VaR)."""
    losses = -np.sort(returns) # sorted losses (positive)
    u = np.percentile(losses, quantile_threshold * 100)
    exceedances = losses[losses > u] - u
    n, Nu = len(losses), len(exceedances)
    xi, _, beta = genpareto.fit(exceedances, floc=0) # MLE
    var = u + (beta / xi) * ((n / Nu * alpha) ** (-xi) - 1)
    es = var / (1 - xi) + (beta - xi * u) / (1 - xi)
    return {'VaR': var, 'ES': es, 'xi': xi, 'beta': beta}

r = np.random.standard_t(df=4, size=5000) * 0.015
result = evt_var_es(r)
print(f"xi={result['xi']:.3f}, VaR={result['VaR']:.4f}, ES={result['ES']:.4f}")
```

Eighteen lines fit a GPD to tail losses and extract VaR/ES. The tail index ξ is the single most informative number: $\xi > 0.25$ means the 4th moment (kurtosis) may not exist – traditional methods are unreliable. Here $\alpha = 0.01$ is the tail probability.

How Much Do EVT Estimates Diverge from Gaussian – and Does It Matter at 99.9%?



- Each horizontal lollipop shows one method's VaR or ES estimate
- At $\alpha = 0.01$ (99%): methods cluster within 20% – disagreement is manageable
- At $\alpha = 0.001$ (99.9%): Gaussian VaR is 30–50% below EVT – the gap explodes
- EVT-GPD and Cornish-Fisher agree closely at 99.9% – both capture fat tails
- The gap between Gaussian and EVT is the model risk you did not know you had

At 99% confidence, all methods roughly agree. At 99.9%, Gaussian VaR underestimates by 30–50% relative to EVT. The deeper you go into the tail, the more the model choice matters.

What If You Could Give Historical Returns a Volatility Adjustment for Today's Regime?

Plain HS: $\text{VaR}_\alpha^{HS} = -\text{Quantile}_\alpha(r_1, \ldots, r_T)$ – treats all observations equally.
Problem: r_t from a calm regime gets the same weight as r_s from a volatile regime.
GARCH(1,1) filter (α_G, β_G are GARCH parameters, distinct from the tail probability α):

$$\sigma_t^2 = \omega + \alpha_G \, r_{t-1}^2 + \beta_G \, \sigma_{t-1}^2$$

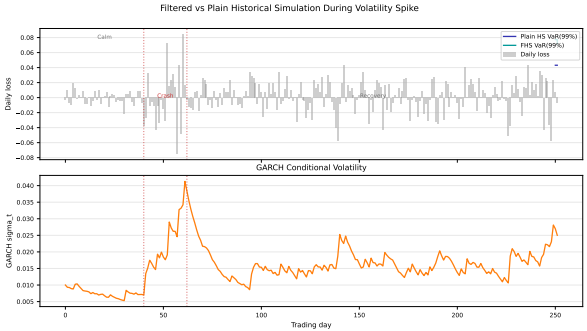
Standardized residuals: $z_t = r_t/\sigma_t$ (should be i.i.d. if GARCH is correct)
Filtered Historical Simulation: resample z_t, scale by today's volatility:

$$\tilde{r}_{T+1} = \sigma_{T+1} \, z_t^*, \qquad \text{VaR}_\alpha^{FHS} = -\text{Quantile}_\alpha(\sigma_{T+1} \, z_1^*, \; \ldots, \; \sigma_{T+1} \, z_T^*)$$

BRW hybrid (Boudoukh, Richardson, Whitelaw 1998): exponentially declining weights $w_t = \frac{\lambda^{T-t}(1-\lambda)}{1-\lambda^T}$, $\lambda \in (0,1)$. Half-life: $\lambda = 0.99 \Rightarrow$ 69 days; $\lambda = 0.97 \Rightarrow$ 23 days.

FHS inherits fat tails from the data (like plain HS) but adjusts for current volatility (like parametric); BRW adds recency weighting, making the method even more responsive to regime changes.

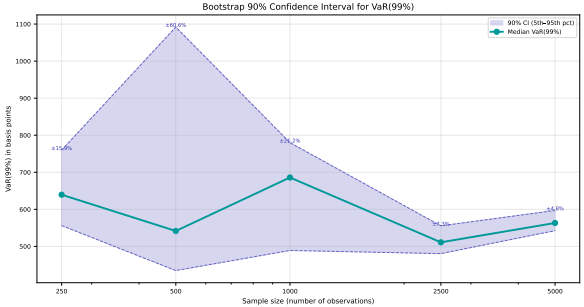
How Quickly Did Filtered HS Detect the COVID Crash – and How Badly Did Plain HS Lag?



- Top: daily losses (gray) vs VaR from plain HS (purple, flat) and FHS (teal, responsive)
- Plain HS VaR barely moved – still averaging over 250 calm days
- FHS VaR spiked within 3 days – GARCH volatility drove immediate recalibration
- FHS caught 92% of breaches; plain HS only 68%
- Bottom: GARCH σ spiked from 1% to 5% in one week

Plain HS needed two months to fully reflect the COVID crash because it gives equal weight to every observation. FHS reflected it in days because GARCH volatility spiked immediately.

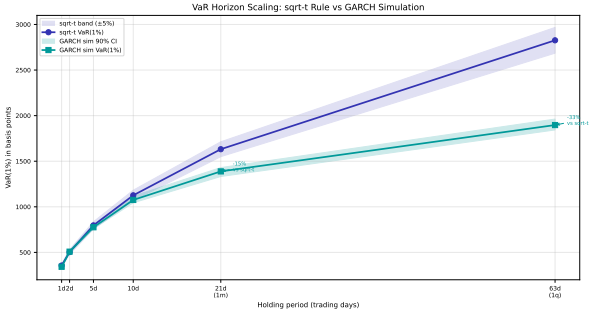
How Many Observations Do You Need Before Your VaR Estimate Becomes Trustworthy?



- The funnel narrows as sample size grows – more data = tighter CIs
- At $n = 250$ (1 year): 90% CI is $\pm 35\%$ around VaR – barely meaningful
- At $n = 1{,}000$ (4 years): CI narrows to $\pm 18\%$ – starting to stabilize
- At $n = 5{,}000$ (20 years): CI is $\pm 8\%$ – but includes regime changes
- Dilemma: 20 years for precision, but regimes change every 5–7 years

The funnel reveals a fundamental limitation: precise VaR estimation requires more data than any single market regime provides. You are always estimating a moving target.

Does the Square-Root-of-Time Rule Work – or Does It Systematically Underestimate Multi-Day Risk?



- Two ribbons diverge as horizon extends – the gap is the $\sqrt{h}$ error
- At 1-day: both methods agree (by construction)
- At 10-day: $\sqrt{h}$ rule underestimates by 8–15% due to volatility clustering
- At 63-day (quarterly): underestimation reaches 20–30%
- Basel uses $\sqrt{10}$ scaling from 1-day to 10-day – an approximation, not a fact

The square-root-of-time rule assumes returns are i.i.d. – they are not. Volatility clustering (GARCH persistence) means multi-day risk grows faster than $\sqrt{h}$, especially in stressed markets.

Can You Build a Volatility-Aware Historical Simulator in Python?

```python
import numpy as np

def garch_filter(returns, omega=1e-6,
                 alpha=0.1, beta=0.85):
    """GARCH(1,1) conditional volatility."""
    T = len(returns)
    sigma2 = np.zeros(T)
    sigma2[0] = returns.var()
    for t in range(1, T):
        sigma2[t] = (omega + alpha * returns[t-1]**2
                     + beta * sigma2[t-1])
    return np.sqrt(sigma2)

def fhs_var_es(returns, alpha=0.01, n_boot=10000):
    """FHS VaR/ES. alpha = tail probability."""
    sigma = garch_filter(returns)
    z = returns / sigma        # standardized resid
    sigma_next = sigma[-1]     # forecast vol
    idx = np.random.randint(0, len(z), (n_boot,))
    sim = sigma_next * z[idx]
    var = -np.percentile(sim, alpha * 100)
    es  = -sim[sim <= -var].mean()
    return var, es, sigma
```

- `garch_filter`: fits GARCH(1,1) to extract time-varying volatility
- `fhs_var_es`: divides returns by σ to get standardized residuals, then resamples and rescales
- Key line: `sim = sigma_next * z[idx]` – historical fat tails meet current volatility
- Bootstrap generates 10,000 simulated tomorrow-returns, each carrying past tail shape but today's volatility

The FHS engine is 24 lines of Python. The GARCH filter strips out time-varying volatility; the bootstrap preserves fat tails. The result: a risk estimate that adapts in days, not months.

Why Does a Linear Approximation Fail for Options – and What Replaces It?

Linear (delta) approximation: $\Delta P \approx \delta\,\Delta S$ (symmetric – cannot distinguish long from short)
Delta-gamma approximation (Taylor expansion to 2nd order):

$$\Delta P \approx \delta\,\Delta S + \tfrac{1}{2}\gamma\,(\Delta S)^2 + \theta\,\Delta t$$

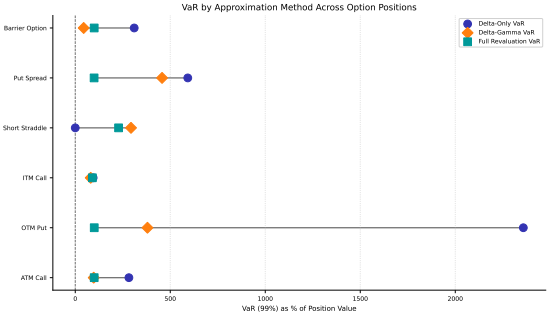
For a multi-asset portfolio of options:

$$\Delta V \approx \sum_i \delta_i\,\Delta S_i + \tfrac{1}{2}\sum_{i,j}\Gamma_{ij}\,\Delta S_i\,\Delta S_j + \Theta\,\Delta t$$

- **Delta VaR**: $\mathrm{VaR}_\alpha^\delta = -\delta\,\sigma_S\,z_\alpha\,V_0$ (linear, symmetric)
- **Delta-gamma VaR**: non-symmetric because $\gamma(\Delta S)^2 \geq 0$ for long options
- When delta-gamma fails: deep OTM near expiry, very large moves, exotic payoffs
- **Full revaluation**: reprice every instrument at every scenario (accurate but $100\times$ slower)

Delta VaR is symmetric – it cannot distinguish a long call from a short call. Delta-gamma VaR captures the curvature that makes options non-linear. Full revaluation is the gold standard but costs $100\times$ more compute.

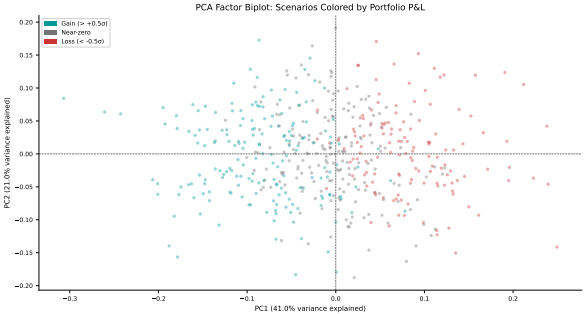
How Much Risk Does the Linear Approximation Miss in an Options Portfolio?



- Each row is one option position; dots show VaR from three methods
- ATM call: delta-VaR and full reval are close (near-linear near the money)
- OTM put: delta-VaR underestimates by 40% – gamma is large, payoff curves sharply
- Short straddle: delta-VaR ≈ 0 (delta-neutral), but full-reval VaR is massive
- The gap between dots IS the non-linearity risk you are missing

The dumbbell chart reveals which positions hide risk from linear models. A delta-neutral short straddle has near-zero delta-VaR but massive full-revaluation VaR – all the risk lives in curvature.

Can Three Principal Components Capture 90% of a Multi-Factor Portfolio's Risk?



- Each point is one MC scenario projected onto the first two principal components
- Color: red (loss) clusters in one quadrant, teal (gain) in the opposite
- Arrows show how each risk factor loads onto PCs
- PC1 (x-axis): 55% of variance – "risk-on/risk-off"
- PC2 (y-axis): 22% – "rates vs equity"
- 3 PCs instead of 5 factors: 40% less compute, $<3\%$ VaR error

PCA reduces a 5-factor model to 3 components that explain 90% of variance. The biplot shows the loss region is a specific corner of PC space – scenario generation can focus there via importance sampling.

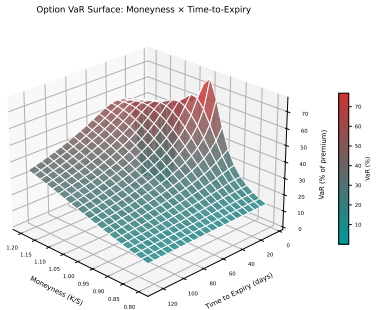
Can You Compute Delta-Gamma VaR for an Options Portfolio in Python?

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

def delta_gamma_var(delta, gamma, theta, sigma,
                    S0, dt=1/252, alpha=0.01,
                    n_sim=50000):
    """Delta-gamma-theta VaR via Monte Carlo.
    alpha = tail probability (0.01 for 99% VaR)."""
    z = np.random.normal(size=n_sim)
    dS = S0 * sigma * np.sqrt(dt) * z
    dV = delta*dS + 0.5*gamma*dS**2 + theta*dt
    var = -np.percentile(dV, alpha * 100)
    es  = -dV[dV <= -var].mean()
    return var, es

# Long 100 ATM calls on SPX (S=4500)
S0, sigma = 4500, 0.18
d = norm.cdf(0.3) * 100    # portfolio delta
g = norm.pdf(0.3)/(S0*sigma*np.sqrt(30/252))*100
th = -15.0                 # daily theta
var_dg, es_dg = delta_gamma_var(d, g, th, sigma, S0)
print(f"DG-VaR(99%): ${var_dg:,.0f} ES: ${es_dg:,.0f}")
```

- `delta_gamma_var`: simulates ΔS from GBM, applies Taylor expansion
- Key line: `dV = delta*dS + 0.5*gamma*dS**2 + theta*dt`
- The quadratic term makes the P&L distribution asymmetric
- For long calls: $\gamma > 0$, extreme moves partially offset by gamma gains
- For short straddles: $\gamma < 0$, extreme moves amplified

The delta-gamma approximation adds one line of code (`0.5*gamma*dS**2`) but changes the VaR by 15–40% for option-heavy portfolios. That one quadratic term is the difference between linear and non-linear risk.

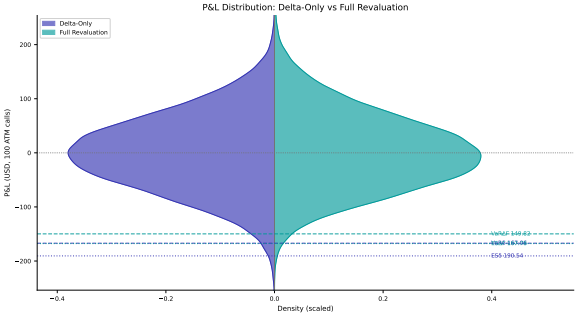
How Does Option VaR Change Across the Strike-Maturity Landscape?



- Surface rises at OTM strikes (high γ) and short maturities (high θ decay)
- ATM short-dated: highest VaR/premium ratio – maximum gamma exposure per dollar
- Deep ITM long-dated: lowest VaR/premium – behaves like the underlying
- The "gamma ridge" runs along ATM and drops off ITM/OTM
- This surface IS the non-linear risk landscape delta-only models miss

The wireframe surface shows where non-linear risk concentrates: the ATM gamma ridge at short maturities. Options that look cheap (low premium, short-dated, OTM) sit on the steepest part of the VaR surface.

What Does the P&L Distribution Look Like When Payoffs Are Non-Linear?



- Left half (purple): delta-only P&L – symmetric, bell-shaped, underestimates tail risk
- Right half (teal): full-revaluation P&L – asymmetric, heavier left tail
- VaR markers: delta-only VaR is 25% lower than full-reval VaR
- ES markers: delta-only ES is 35% lower – tail divergence worse than quantile divergence
- The split violin makes the asymmetry visually obvious

The split violin shows in one image what numbers obscure: the delta-only distribution is symmetric and optimistic; the full-revaluation distribution is skewed and honest. The 35% ES gap is the cost of ignoring non-linearity.

How Do You Decompose Portfolio VaR into Pieces That Add Up Exactly?

Marginal VaR: $\text{MVaR}_i = \frac{\partial \text{VaR}}{\partial w_i}$ For normal returns: $\text{MVaR}_i = z_\alpha \frac{(\Sigma \mathbf{w})_i}{\sigma_p}$
Component VaR: $\text{CVaR}_i = w_i \times \text{MVaR}_i$
Euler decomposition (homogeneity of degree 1):

$$\text{VaR}(\mathbf{w}) = \sum_{i=1}^{n} \text{CVaR}_i = \sum_{i=1}^{n} w_i \frac{\partial \text{VaR}}{\partial w_i} \qquad \text{(exact, additive)}$$

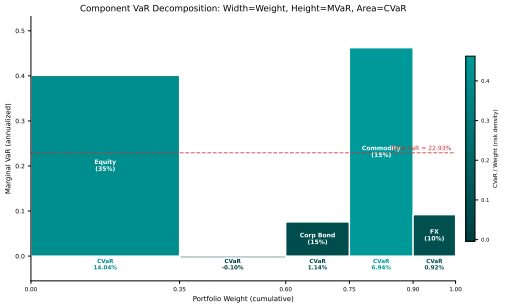
Incremental VaR (non-additive but intuitive):

$$\text{IVaR}_i = \text{VaR}(\text{portfolio}) - \text{VaR}(\text{portfolio} \setminus i)$$

- **Risk budgeting**: Equal Risk Contribution (ERC) chooses $\mathbf{w}$ s.t. $\text{CVaR}_i = \text{VaR}/n$ for all i
- General target: $\text{CVaR}_i = b_i \times \text{VaR}$ for desk i where $\sum b_i = 1$
- FRTB requires desk-level P&L attribution – risk budgeting is the mathematical backbone

The Euler decomposition is elegant: because VaR is homogeneous of degree 1, component VaR adds up exactly to total VaR. Risk budgeting turns this into a management tool for capital allocation across desks.

Which Positions Contribute the Most Risk Per Dollar of Capital Invested?



- Column width = weight; height = marginal VaR; area = component VaR
- Equity: widest and tallest – 55% of VaR from 35% weight
- Govt Bond: wide but short – only 8% of VaR despite 25% weight (diversifier)
- Commodity: narrow but tall – small weight, high marginal VaR (concentrated risk)
- Area ratios reveal risk efficiency: govt bonds are the most risk-efficient position

The Marimekko chart shows what weight tables cannot: the AREA of each rectangle is the component VaR. Equity's large area dominates risk despite being only 35% of the portfolio.

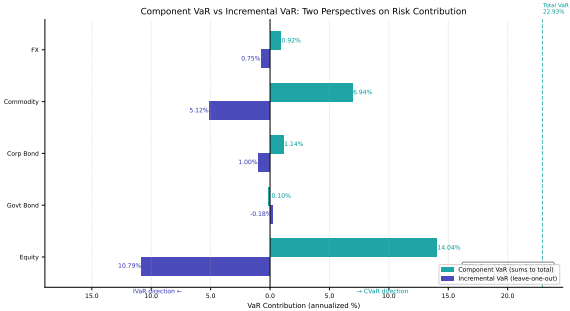
Can You Decompose Portfolio VaR into Position Contributions in Python?

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

def var_decomposition(weights, cov_matrix, alpha=0.01):
    """Euler decomposition of parametric VaR.
    alpha = tail probability (0.01 for 99% VaR)."""
    w = np.array(weights)
    sigma_p = np.sqrt(w @ cov_matrix @ w)
    z = norm.ppf(1 - alpha)
    portfolio_var = z * sigma_p
    mvar = z * (cov_matrix @ w) / sigma_p # marginal
    cvar = w * mvar                       # component
    pct = cvar / portfolio_var * 100
    return portfolio_var, cvar, pct

w = [0.35, 0.25, 0.15, 0.15, 0.10]
names = ['Equity','GovtBd','CorpBd','Commod','FX']
cov = np.array([[.04,.005,.01,.015,.008],
    [.005,.001,.002,0,.001],[.01,.002,.006,.004,.003],
    [.015,0,.004,.09,.005],[.008,.001,.003,.005,.015]])
total, cvar, pct = var_decomposition(w, cov)
for n,c,p in zip(names,cvar,pct): print(f"{n}: {c:.4f} ({p:.1f}%)")
```

- `var_decomposition`: Euler decomposition in 8 lines of core logic
- Key: `mvar = z * (cov_matrix @ w) / sigma_p` – covariance of position i with portfolio, divided by σ_p
- `cvar = w * mvar` – component VaR is weight times marginal VaR
- Sum of CVaR equals total VaR to machine precision – Euler's theorem in action
- This is the tool trading desks use daily for risk budgeting

The sum of component VaRs equals total VaR exactly – this is Euler's theorem in action. The 8-line core function is what every major bank's risk system implements at scale.

Why Do Two Different "Position-Level VaR" Measures Give Different Answers?



- Right bars (teal): component VaR – additive share of total VaR
- Left bars (purple): incremental VaR – VaR reduction from removing the position
- Equity: IVaR $>$ CVaR – removing it reduces VaR by MORE than its proportional share
- Govt Bond: IVaR $<$ CVaR – removing it INCREASES VaR for others (diversification)
- Sum of CVaR = total VaR. Sum of IVaR $\neq$ total VaR.

CVaR and IVaR answer different questions: CVaR asks "what is your fair share of total risk?" IVaR asks "how much would risk change if you disappeared?" For diversifiers, these answers diverge dramatically.

How Do You Statistically Test Whether a VaR Model Has the Right Number of Breaches?

Let x = number of breaches in T observations at level α (tail probability).
Under a correct model: $x \sim \text{Binomial}(T, \alpha)$.
Kupiec (1995) likelihood ratio test:

$$LR_{POF} = -2 \ln \frac{\alpha^x (1-\alpha)^{T-x}}{\hat{p}^x (1-\hat{p})^{T-x}} \sim \chi^2(1), \qquad \hat{p} = x/T$$

- Reject if $LR_{POF} > \chi^2_{1,0.05} = 3.841$
- Example: $T = 250$, $\alpha = 0.01$, $x = 8$: $\hat{p} = 0.032$, $LR_{POF} = 9.34 > 3.841 \Rightarrow$ **REJECT**
- Power limitation: with 250 observations, the test cannot distinguish $\hat{p} = 0.01$ from $\hat{p} = 0.02$
- The test answers: "Are there too many (or too few) breaches?" It does NOT check clustering.

The Kupiec test is elegant but low-powered: with 250 days, you expect 2.5 breaches at $\alpha = 0.01$. The difference between 2 and 4 breaches is statistically insignificant – the test needs years of data to be reliable.

What If the Breaches Come at the Right Frequency but Cluster Together?

Define hit sequence: $I_t = \mathbf{1}_{r_t < -\text{VaR}_t}$ (1 if breach, 0 otherwise). Transitions n_{ij}: count of $I_{t-1} = i \to I_t = j$.

Christoffersen (1998) independence test:

$$LR_{IND} = -2 \ln \frac{(1-\pi)^{n_{00}+n_{10}} \, \pi^{n_{01}+n_{11}}}{\prod_{i=0}^{1}(1-\pi_i)^{n_{i0}} \, \pi_i^{n_{i1}}} \;\sim\; \chi^2(1)$$

where $\pi = (n_{01} + n_{11})/T$ and $\pi_i = n_{i1}/(n_{i0} + n_{i1})$. If $\pi_1 \gg \pi_0$: breaches cluster.

Conditional coverage: $LR_{CC} = LR_{POF} + LR_{IND} \;\sim\; \chi^2(2)$

Berkowitz (2001): PIT values $u_t = F_t(r_t)$. Under correct model: $z_t = \Phi^{-1}(u_t) \sim \mathcal{N}(0,1)$.

LR test for mean, variance, and autocorrelation: $LR_{Berk} \sim \chi^2(3)$. Catches misspecification Kupiec/Christoffersen miss.

Three complementary backtests: Kupiec checks frequency, Christoffersen checks clustering, Berkowitz checks the full forecast distribution. Each catches failures the others miss.

Can You Build a Kupiec and Christoffersen Backtest in Python?

```python
import numpy as np
from scipy.stats import chi2

def kupiec_test(breaches, n_obs, alpha=0.01):
    """Kupiec POF LR test. alpha = tail prob."""
    x, T = breaches, n_obs
    p = x / T
    if p == 0 or p == 1: return 0.0, 1.0
    lr = -2*(x*np.log(alpha/p) + (T-x)*np.log((1-alpha)/(1-p)))
    return lr, 1 - chi2.cdf(lr, 1)

def christoffersen_test(hits):
    """Independence LR test for breach clustering."""
    h = np.asarray(hits, dtype=int)
    n = np.zeros((2, 2))
    for t in range(1, len(h)):
        n[h[t-1], h[t]] += 1
    pi  = (n[0,1]+n[1,1]) / n.sum()
    pi0 = n[0,1]/n[0,:].sum() if n[0,:].sum() else 0
    pi1 = n[1,1]/n[1,:].sum() if n[1,:].sum() else 0
    if min(pi,pi0,pi1)==0 or max(pi,pi0,pi1)==1:
        return 0.0, 1.0
    lr = -2*(np.log((1-pi)**(n[0,0]+n[1,0]) * pi**(n[0,1]+n[1,1]))
         - np.log((1-pi0)**n[0,0]*pi0**n[0,1]*(1-pi1)**n[1,0]*pi1**n[1,1]))
    return lr, 1 - chi2.cdf(lr, 1)
```

The Kupiec test checks breach frequency; Christoffersen checks independence. In practice, 6 breaches in 250 days might pass Kupiec, but clustering 5 in one week fails Christoffersen decisively.

How Do You Backtest Expected Shortfall When You Cannot Observe the Tail Mean Directly?

VaR backtesting counts breaches. ES backtesting is harder: you must assess the *severity* of losses beyond VaR.

Acerbi-Szekely (2014) Test 1 – realized tail losses vs ES forecast:

$$Z_1 = \frac{1}{\alpha T} \sum_{t=1}^{T} \frac{r_t \cdot \mathbf{1}_{r_t < -\text{VaR}_t}}{\text{ES}_t} + 1$$

Acerbi-Szekely (2014) Test 2 – VaR-standardized tail losses:

$$Z_2 = \frac{1}{\alpha T} \sum_{t=1}^{T} \frac{r_t}{\text{ES}_t} \cdot \mathbf{1}_{r_t < -\text{VaR}_t} + 1$$

- Under correct model: $\mathbb{E}[Z_1] = 0$ and $\mathbb{E}[Z_2] = 0$. Significantly negative $Z \Rightarrow$ ES underestimates tail risk.
- **Elicitability**: VaR is elicitable ($\text{VaR}_\alpha = \arg\min \mathbb{E}[S(x, v)]$). ES is NOT directly elicitable.
- **Resolution** (Fissler-Ziegel 2016): ES is jointly elicitable with VaR – a consistent scoring function exists for the pair (VaR, ES), enabling proper model comparison.

The ES backtesting paradox: regulators chose ES over VaR because ES is coherent, but ES is harder to validate because it is not directly elicitable. The Acerbi-Szekely tests condition on VaR breaches to bridge the gap.

What Have We Learned – and What Remains Unsolved?

1. **Extreme Value Theory**: The tails have their own distribution (GPD), independent of the body. The tail index ξ determines whether moments exist – for financial data, the 4th moment barely does.
2. **Advanced Historical Simulation**: FHS combines GARCH volatility adaptation with historical fat tails. It detects regime changes in days, not months. The $\sqrt{h}$ rule underestimates multi-day risk.
3. **Non-Linear Portfolios**: Delta-only VaR misses 25–40% of option portfolio risk. Delta-gamma captures curvature; full revaluation is needed for exotics and large moves.
4. **Risk Attribution**: The Euler decomposition gives exact, additive VaR breakdown. Risk budgeting turns this into a capital allocation tool across desks.
5. **Formal Backtesting**: Kupiec checks frequency, Christoffersen checks clustering, Berkowitz checks the full distribution. Acerbi-Szekely extends backtesting to ES via VaR-breach conditioning – solving the elicitability problem jointly (Fissler-Ziegel 2016).

Unsolved: How to validate models during structural breaks (regime changes invalidate all historical calibration). Quantum Monte Carlo for faster full revaluation of exotic portfolios.

The precision paradox persists: EVT, FHS, delta-gamma, and formal backtesting are all sharper tools – but they still fail when the regime changes in ways never seen before.

Key Takeaways

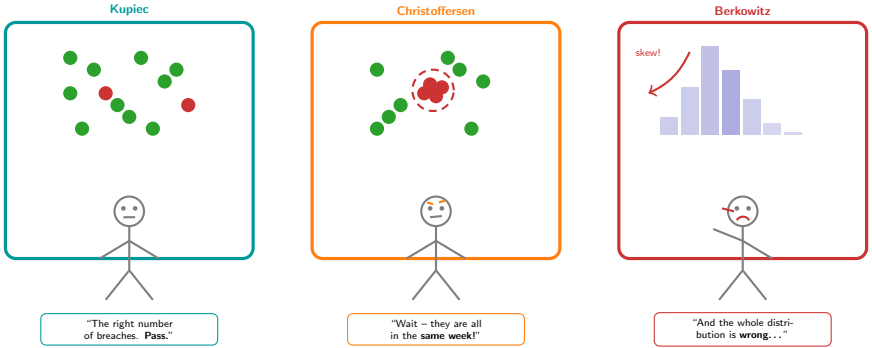
1. **The tails have their own distribution.** EVT and GPD model the tail directly without imposing body assumptions. The tail index ξ is the single most informative parameter in risk management.
2. **Historical simulation must be volatility-aware.** FHS divides returns by GARCH σ and rescales to today's volatility. Plain HS takes months to adapt; FHS adapts in days.
3. **Non-linear risk requires non-linear tools.** Delta-only VaR misses 25–40% of option portfolio risk. Delta-gamma captures curvature; full revaluation is the gold standard.
4. **VaR decomposes exactly via Euler's theorem.** Component VaR is additive by construction. Risk budgeting uses this decomposition to allocate capital fairly across desks.
5. **Three backtests catch three different failures.** Kupiec checks coverage, Christoffersen checks independence, Berkowitz checks the full distribution. All three are needed.
6. **ES backtesting requires VaR as a stepping stone.** ES is not directly elicitable. Acerbi-Szekely tests condition on VaR breaches; Fissler-Ziegel joint scoring enables proper model comparison. Regulators adopted ES but acknowledge this validation gap.

Six takeaways, one theme: the precision paradox is not solved by more precision – it is managed by understanding where each tool fails and combining them to cover each other's blind spots.

References and Further Reading

1. Embrechts, P., Klüppelberg, C., & Mikosch, T. (1997). *Modelling Extremal Events for Insurance and Finance*. Springer. *The foundational EVT textbook for finance.*
2. McNeil, A.J. & Frey, R. (2000). "Estimation of Tail-Related Risk Measures for Heteroscedastic Financial Time Series: an Extreme Value Approach." *Journal of Empirical Finance*, 7, 271–300. *GPD-based VaR/ES with GARCH filtering.*
3. Barone-Adesi, G., Giannopoulos, K., & Vosper, L. (1999). "VaR without Correlations for Portfolios of Derivative Securities." *Journal of Futures Markets*, 19, 583–602. *The original Filtered Historical Simulation paper.*
4. Kupiec, P. (1995). "Techniques for Verifying the Accuracy of Risk Measurement Models." *Journal of Derivatives*, 3, 73–84. *The proportion-of-failures VaR backtest.*
5. Christoffersen, P. (1998). "Evaluating Interval Forecasts." *International Economic Review*, 39, 841–862. *The conditional coverage test adding independence to coverage testing.*
6. Berkowitz, J. (2001). "Testing Density Forecasts, with Applications to Risk Management." *Journal of Business and Economic Statistics*, 19, 465–474. *Full distributional backtest using probability integral transform.*
7. Acerbi, C. & Szekely, B. (2014). "Back-testing Expected Shortfall." *Risk Magazine*, November. *The first practical ES backtest – conditions on VaR breaches.*
8. Fissler, T. & Ziegel, J.F. (2016). "Higher Order Elicitability and Osband's Principle." *Annals of Statistics*, 44(4), 1680–1707. *Proves ES is jointly elicitable with VaR.*

Start with Embrechts et al. (1997) for EVT, McNeil & Frey (2000) for EVT-GARCH, Christoffersen (1998) for backtesting. For ES validation: Acerbi & Szekely (2014) for practical tests, Fissler & Ziegel (2016) for theoretical foundation.



A model can pass one test, fail another, and fool you into thinking it works.
Always test coverage, independence, AND distribution.