L03: Robo-Advisory & Algorithmic Wealth Management

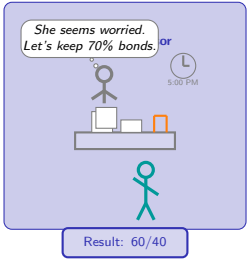
Extended Slides – BSc Digital Finance Course

Digital Finance

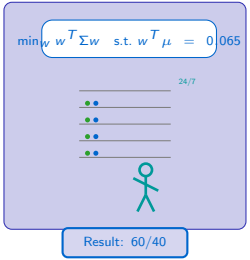
What Will You Be Able to Do After This Lecture?

1. Derive the mean-variance optimization problem and solve it analytically using Lagrange multipliers
2. Implement Black-Litterman portfolio allocation in Python and compare posteriors with Markowitz priors
3. Design and evaluate rebalancing algorithms with threshold-based and calendar-based triggers
4. Quantify tax-loss harvesting value using after-tax return simulations
5. Model behavioral finance biases mathematically and measure their portfolio impact
6. Evaluate ESG integration strategies in robo-advisory and their risk-return trade-offs

Six objectives: formal optimization (1), Bayesian allocation (2), algorithm design (3–4), behavioral economics (5), and sustainable investing (6).



"Are you sure about this?"



Rebalanced. No action needed.

Same portfolio. Different bedside manner. Which one do you trust at 3am?

Why Is the Efficient Frontier Shaped Like a Parabola?

Constrained optimization: $\min_w \frac{1}{2} w^T \Sigma w \quad \text{s.t. } w^T \mu = \mu_p, \quad w^T \mathbf{1} = 1$
Lagrangian:

$$\mathcal{L} = \frac{1}{2} w^T \Sigma w - \lambda_1 (w^T \mu - \mu_p) - \lambda_2 (w^T \mathbf{1} - 1)$$

First-order condition: $\Sigma w = \lambda_1 \mu + \lambda_2 \mathbf{1} \quad \Rightarrow \quad w^* = \Sigma^{-1}(\lambda_1 \mu + \lambda_2 \mathbf{1})$
Define scalars: $A = \mathbf{1}^T \Sigma^{-1} \mu, \quad B = \mu^T \Sigma^{-1} \mu, \quad C = \mathbf{1}^T \Sigma^{-1} \mathbf{1}, \quad D = BC - A^2$
Frontier equation (parabola in (σ_p, μ_p) space):

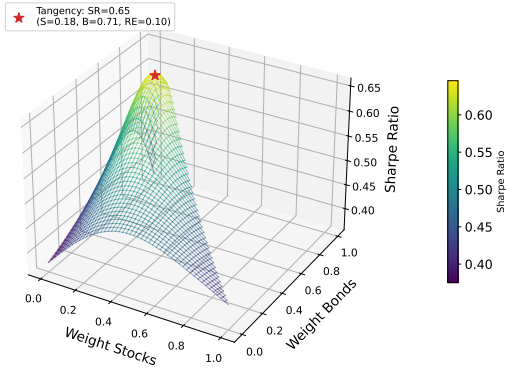
$$\sigma_p^2 = \frac{C \mu_p^2 - 2A\mu_p + B}{D}$$

Global minimum variance portfolio: $w_{GMV} = \dfrac{\Sigma^{-1} \mathbf{1}}{\mathbf{1}^T \Sigma^{-1} \mathbf{1}}$

The efficient frontier is a parabola because portfolio variance is a quadratic function of target return. This is not an approximation – it is exact under Markowitz assumptions.

What Does the Efficient Frontier Look Like in Three Dimensions?

Efficient Frontier: 3-Asset Sharpe Surface



- Wireframe surface: x = weight in stocks, y = weight in bonds, z = Sharpe ratio
- The "ridge" along the surface is the efficient frontier projected into weight space
- Color encodes Sharpe ratio: warm = high, cool = low
- Tangency portfolio sits at the ridge peak – highest Sharpe ratio achievable
- Most weight combinations cluster far from the ridge: the vast majority of portfolios are suboptimal
- The 3D view reveals what 2D frontier plots hide: the Sharpe surface is smooth but has a single narrow ridge

The wireframe reveals what 2D efficient frontier plots hide: the Sharpe surface is smooth but has a single narrow ridge. Most weight combinations are suboptimal.

Can You Solve Markowitz Optimization in 18 Lines?

```python
import numpy as np
from scipy.optimize import minimize

def markowitz(mu, cov, target_ret, rf=0.015):
    """Min-variance portfolio for a target return."""
    n = len(mu)
    def objective(w):
        return w @ cov @ w
    constraints = [
        {'type': 'eq', 'fun': lambda w: w @ mu - target_ret},
        {'type': 'eq', 'fun': lambda w: np.sum(w) - 1}]
    bounds = [(0, 1)] * n
    res = minimize(objective, np.ones(n)/n, method='SLSQP',
                   bounds=bounds, constraints=constraints)
    vol = np.sqrt(res.fun)
    sharpe = (target_ret - rf) / vol
    return res.x, vol, sharpe
```

This 18-line function is the engine inside every robo-advisor. Production systems add transaction costs, tax constraints, and sector limits – but the core math is identical.

Why Is Variance a Terrible Measure of Investment Risk?

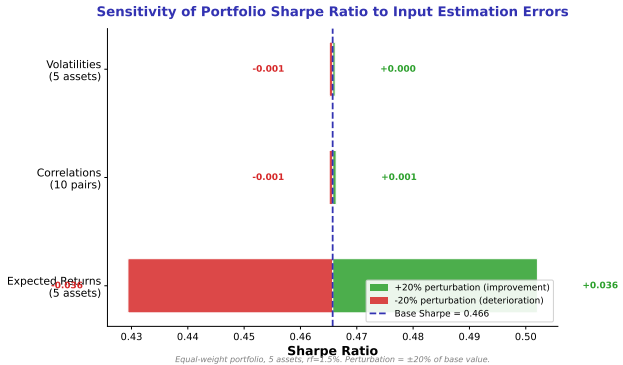
Variance treats upside and downside equally: $\sigma^2 = E[(R-\mu)^2]$
Semi-variance (downside only): $\sigma_d^2 = E[\min(R-\mu, 0)^2]$
Value-at-Risk: $P(R < -\text{VaR}_\alpha) = \alpha$. For normal returns: $\text{VaR}_\alpha = \mu - z_\alpha \sigma$
Conditional VaR (Expected Shortfall): $\text{CVaR}_\alpha = E[R \mid R < -\text{VaR}_\alpha]$
Maximum Drawdown:

$$\text{MDD} = \max_{t \in [0,T]} \left(\frac{\max_{s \in [0,t]} P_s - P_t}{\max_{s \in [0,t]} P_s} \right)$$

Sortino ratio: $\dfrac{R_p - r_f}{\sigma_d}$ (penalizes only downside volatility)
Which matters for retirees? MDD – because they cannot wait for recovery.

Variance is symmetric. Investors are not. A retiree who loses 30% cannot wait 5 years for recovery – MDD and CVaR capture this asymmetry.

Which Input Error Hurts Your Portfolio the Most?



- Tornado chart: sensitivity of portfolio Sharpe ratio to $\pm 20\%$ estimation error in each input
- Expected returns dominate (widest bar) – errors here swing Sharpe by 5x more than covariance errors
- Correlations matter moderately; variances matter least
- This is why Black-Litterman exists: stabilize the most sensitive input (expected returns) by anchoring to equilibrium
- Key insight: "garbage in, garbage out" is not symmetric – return errors matter 5x more than covariance errors

Expected return estimation errors dominate all others. This single fact motivates Black-Litterman, risk parity, and every robust optimization technique.

How Do You Combine Market Consensus with Your Own Views?

Equilibrium returns (reverse optimization): $\Pi = \delta \Sigma w_{mkt}$ where δ = risk aversion, w_{mkt} = market-cap weights
Investor views: $P\mu = Q + \epsilon, \quad \epsilon \sim N(0, \Omega)$
Posterior expected returns:

$$\hat{\mu} = \left[(\tau\Sigma)^{-1} + P^T\Omega^{-1}P\right]^{-1}\left[(\tau\Sigma)^{-1}\Pi + P^T\Omega^{-1}Q\right]$$

Posterior covariance: $\hat{\Sigma}_\mu = \left[(\tau\Sigma)^{-1} + P^T\Omega^{-1}P\right]^{-1}$
Uncertainty scalar: τ typically 0.025–0.05; small τ = trust the market more
Key insight: With NO views ($P = \emptyset$), BL returns to market-cap weighting – the most diversified possible starting point.

Black-Litterman starts from "the market is right" and lets you tilt toward your views. With no views, you get the market portfolio – the safest possible default.

Can You Implement Black-Litterman in Python?

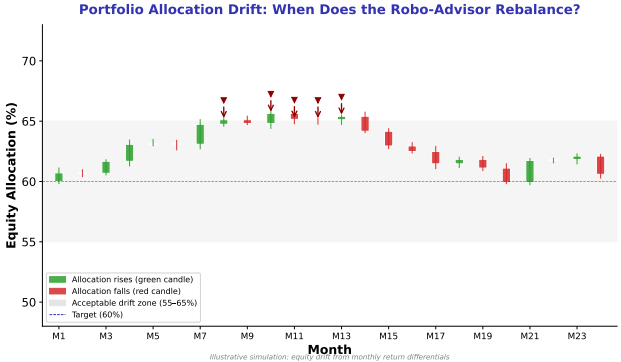
```python
import numpy as np

def black_litterman(cov, mkt_w, views_P, views_Q,
                    tau=0.05, delta=2.5, omega_scale=0.1):
    """Posterior returns blending market equilibrium + views."""
    pi = delta * cov @ mkt_w       # equilibrium returns
    tau_cov_inv = np.linalg.inv(tau * cov)
    omega = omega_scale * np.diag(views_P @ cov @ views_P.T)
    omega = np.diag(omega)
    omega_inv = np.linalg.inv(omega)
    # Posterior return (combined estimator)
    M = np.linalg.inv(tau_cov_inv + views_P.T @ omega_inv @ views_P)
    mu_post = M @ (tau_cov_inv @ pi + views_P.T @ omega_inv @ views_Q)
    return mu_post, M

# Usage: cov=3x3, mkt_w=[0.6,0.3,0.1], P=[[1,-1,0]], Q=[0.02]
mu_post, sigma_post = black_litterman(cov, mkt_w, P, Q)
```

This 18-line function powers the allocation engine of sophisticated robo-advisors like Scalable Capital and Wealthfront. The key parameter is tau: lower = trust the market more.

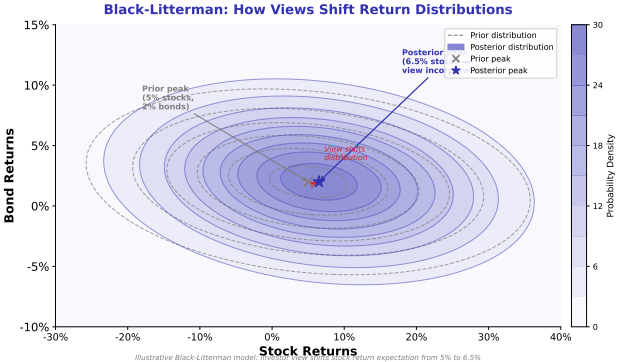
How Does a Portfolio Drift Between Rebalancing Events?



- Candlestick-style chart: monthly portfolio allocation drift over 24 months
- Each "candle": open = start-of-month stock allocation, close = end-of-month, high/low = intra-month extremes
- Red candles = drift toward overweight stocks; green = toward underweight
- Horizontal band at 55–65% marks the acceptable drift zone
- Rebalance triggers (arrows) fire when candle exits the band
- Shows 4–5 rebalance events – too frequent wastes costs, too infrequent increases risk drift

Portfolio drift is inevitable. The question is when to rebalance: too frequent = transaction costs and taxes; too infrequent = risk drift. The threshold band is the engineering compromise.

How Do Investor Views Shift the Return Distribution?



Illustrative Black-Litterman model showing how views shift return expectation from 5% to 6.5%

- Contour overlay: bivariate return distributions for stocks and bonds
- Gray contours = prior (equilibrium returns)
- Colored contours = posterior (after view "stocks outperform bonds by 2%")
- The posterior shifts northeast (higher stock return expectation) and slightly narrows (view reduces uncertainty)
- Shift magnitude depends on view confidence (Ω): high confidence = large shift, low confidence = small
- With zero-confidence views, posterior collapses back to prior – the market wins

Black-Litterman visually: the posterior (colored) shifts toward your view but is anchored by the prior (gray). Uncertain views barely move the needle.

What If You Equalize Risk Contribution Instead of Capital?

Risk contribution of asset i: $RC_i = w_i \cdot (\Sigma w)_i$
Total risk: $\sigma_p = \sqrt{w^T \Sigma w} = \sum_i RC_i$
Risk parity condition: $RC_i = \dfrac{\sigma_p}{n}$ for all i
Equivalent optimization:

$$\min_w \sum_{i=1}^{n} \left(RC_i - \frac{\sigma_p}{n} \right)^2 \quad \text{s.t.} \quad w^T \mathbf{1} = 1, \quad w_i > 0$$

For uncorrelated assets: $w_i \propto \frac{1}{\sigma_i}$ (inverse volatility weighting)
Bridgewater All Weather Fund: Most famous risk parity implementation (∼$150B AUM)
Key advantage: No expected return inputs needed – avoids the estimation error problem entirely
Trade-off: Leveraged bond positions required to match equity returns

Risk parity eliminates the most dangerous input (expected returns) by focusing only on risk balance. The trade-off: it requires leverage to deliver competitive returns.

How Often Should You Rebalance – and Can Math Answer This?

Calendar rebalancing: Fixed interval (monthly, quarterly, annually)
Threshold rebalancing: Trigger when $|w_i - w_i^*| > \delta$ for any asset i
Optimal threshold (Sun et al. 2006): $\delta^* \approx \left(\frac{3c}{4\sigma^2}\right)^{1/3}$ where c = transaction cost, σ = volatility
Expected rebalances per year: $N \approx \frac{\sigma}{\delta}\sqrt{\frac{2T}{\pi}}$
After-tax return: $R_{after} = R_{gross} - f_{mgmt} - c \cdot N - \tau \cdot \text{realized gains}$
Tax-loss harvesting value: $V_{TLH} = \sum_t \tau_{cap} \cdot L_t \cdot (1+r)^{T-t}$ where L_t = harvested loss at time t
Wash-sale constraint: Cannot repurchase "substantially identical" security within 30 days

---

The optimal rebalancing threshold balances three forces: transaction costs (too frequent), risk drift (too infrequent), and tax drag (realized gains). The math gives a clean answer.

Can You Build a Tax-Aware Rebalancing Engine?

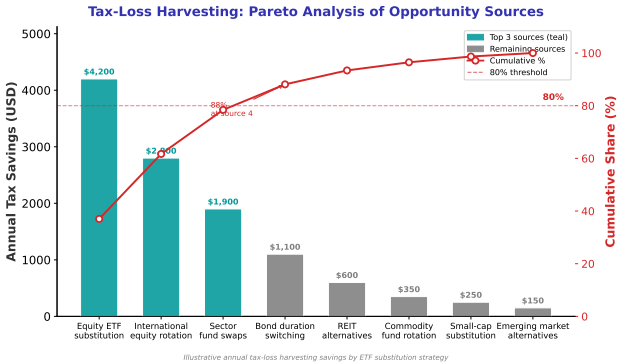
```python
import numpy as np

def tax_aware_rebalance(curr_w, target_w, prices, cost_basis,
                        threshold=0.05, tax_rate=0.25):
    """Rebalance with tax-loss harvesting awareness."""
    drift = np.abs(curr_w - target_w)
    if drift.max() < threshold:
        return {}, 0.0 # no rebalance needed
    trades = target_w - curr_w # positive = buy
    gains = prices - cost_basis
    tlh_value = 0.0
    for i, (trade, gain) in enumerate(zip(trades, gains)):
        if trade < 0 and gain < 0: # selling at a loss
            tlh_value += abs(trade) * abs(gain) * tax_rate
    return dict(enumerate(np.round(trades, 4))), tlh_value

# Usage: curr=[0.68,0.22,0.10], target=[0.60,0.30,0.10]
trades, tlh = tax_aware_rebalance(curr_w, target_w, prices, basis)
```

Tax-loss harvesting identifies positions you are already selling at a loss. The tax savings compound for decades – estimated 0.5–1.5% annually in after-tax alpha.

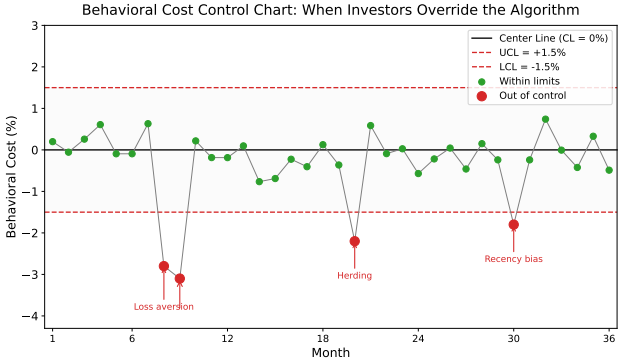
Which Tax-Loss Harvesting Opportunities Matter Most?



- Pareto chart ranking TLH sources by annual tax savings
- Bars (left axis): dollar savings per source – equity ETF substitution leads at $4,200
- Cumulative percentage line (right axis) rises steeply through the first three sources
- Top 3 sources (equity ETF, international rotation, sector swaps) account for ∼80% of all TLH value
- Below the 80% line: diminishing returns from exotic harvesting strategies
- The Pareto principle applies cleanly: complexity beyond the top 3 adds marginal value

The Pareto principle applies to tax-loss harvesting: 80% of the savings come from 3 simple strategies. Exotic harvesting adds complexity but diminishing value.

When Do Behavioral Biases Push Portfolio Decisions Out of Control?



- Shewhart-style control chart: monthly "behavioral cost" (deviation of actual vs. algorithm-recommended returns) over 36 months
- Center line (CL) at 0%, UCL at $+1.5\%$, LCL at -1.5%
- Green points within limits; red points breach control limits during crises
- March 2020 (pandemic panic): loss aversion drives -2.8% deviation
- Feb 2022 (Ukraine invasion): herding pushes -2.2%
- Oct 2023 (rate fears): recency bias causes -1.8%
- Key pattern: behavioral costs are near zero in calm markets but spike during crises

The control chart reveals a pattern: behavioral costs are nearly zero in calm markets but spike out of control during crises. This is exactly when the robo-advisor's discipline matters most.

Why Do Investors Feel Losses Twice as Much as Gains?

Expected Utility Theory: $U = \sum_i p_i \cdot u(x_i)$ where u is concave (risk aversion)
Prospect Theory (Kahneman & Tversky 1979): Value function

$$v(x) = \begin{cases} x^{\alpha} & x \geq 0 \\ -\lambda(-x)^{\beta} & x < 0 \end{cases}$$

Empirical estimates: $\alpha \approx 0.88$, $\beta \approx 0.88$, $\lambda \approx 2.25$ (loss aversion coefficient)
Probability weighting: $w(p) = \dfrac{p^{\gamma}}{(p^{\gamma} + (1-p)^{\gamma})^{1/\gamma}}$ with $\gamma \approx 0.61$
Overweight small probabilities (lottery effect), underweight moderate probabilities
Portfolio implication: Loss-averse investors hold too little equity (willingness to accept volatility is lower than MPT assumes)
Behavioral risk tolerance: $\theta_{behavioral} = \theta_{stated} \cdot \frac{1}{\lambda}$ (approximately half of what they report)

Loss aversion coefficient $\lambda = 2.25$ means a $100 loss hurts as much as a $225 gain. This single parameter explains most investor irrationality.

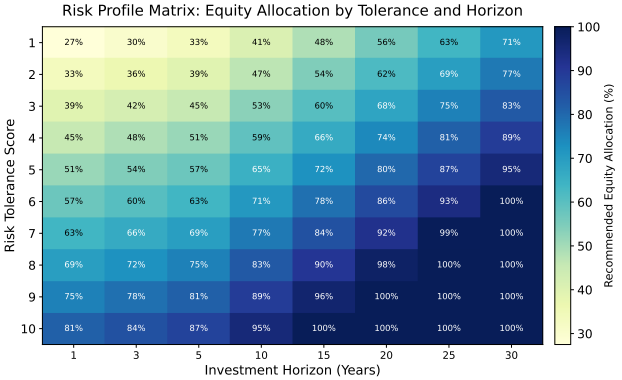
Can You Build a Risk Profiler That Accounts for Behavioral Bias?

```python
import numpy as np

def risk_profile(answers, loss_aversion=2.25):
    """Map questionnaire to risk score, adjusted for bias.
    answers: dict with keys 'horizon', 'loss_react', 'experience',
    'income_stability', 'goal_urgency' each scored 1-5."""
    raw = (answers['horizon'] * 0.30
         + answers['loss_react'] * 0.25
         + answers['experience'] * 0.20
         + answers['income_stability'] * 0.15
         + answers['goal_urgency'] * 0.10)
    raw_pct = (raw - 1) / 4 * 100 # scale to 0-100
    # Adjust for stated vs revealed preference gap
    adjusted = raw_pct / (loss_aversion ** 0.3)
    bucket = ('Conservative' if adjusted < 30
              else 'Moderate' if adjusted < 60
              else 'Aggressive')
    return {'raw': raw_pct, 'adjusted': adjusted, 'bucket': bucket}
```

The adjustment factor ($\lambda^{0.3}$) shrinks the stated risk score by $\sim$25%. This accounts for the gap between what investors SAY they can tolerate and what they ACTUALLY tolerate.

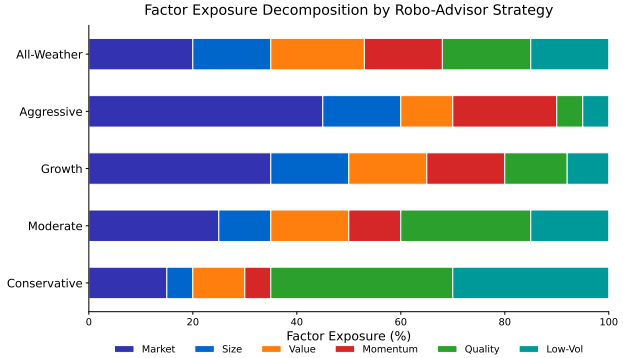
What Asset Mix Does Each Risk Profile and Time Horizon Demand?



- Bivariate heatmap: x = time horizon (1–30 years), y = risk tolerance (1–10), cell color = recommended equity allocation (0–100%)
- Top-right (high risk + long horizon) = 90%+ equity (dark blue)
- Bottom-left (low risk + short horizon) = 10% equity (light yellow)
- The diagonal reveals the "sweet spot" where most robo-advisor clients fall (40–70% equity)
- Critical: the 55–65 year old retiree zone sits bottom-center – low equity regardless of stated risk tolerance
- Most variation comes from TIME HORIZON, not risk tolerance

The heatmap reveals what risk profiling actually does: it maps a 2D space (tolerance $\times$ horizon) to a 1D output (equity allocation). Most of the variation comes from TIME HORIZON, not risk tolerance.

How Do Factor Strategies Differ Inside a Robo-Advisor?



- 100% stacked bar: factor exposure decomposition for 5 robo-advisor strategies
- Factors: Market (β), Size (SMB), Value (HML), Momentum (WML), Quality, Low-Volatility
- Conservative = dominated by low-vol and quality (65% combined)
- Aggressive = dominated by market beta and momentum (65% combined)
- All-Weather = roughly equal factor weights (risk parity in factor space)
- Key insight: what robo-advisors call "risk profiles" are really "factor bets"

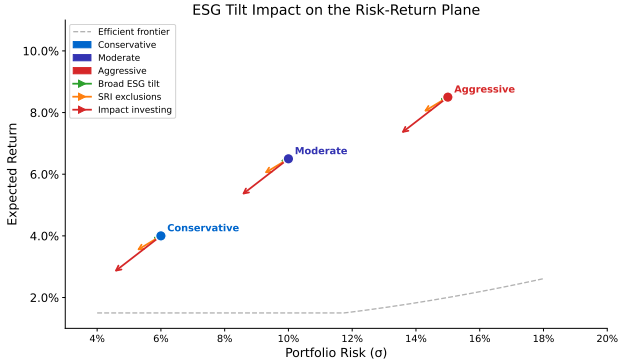
Risk profiles are factor bets in disguise. "Conservative" means "overweight quality and low-vol." "Aggressive" means "overweight beta and momentum." Understanding factors demystifies the algorithm.

Does Adding an ESG Constraint Always Cost You Returns?

Standard Markowitz: $\min_w w^T \Sigma w \quad \text{s.t. } w^T \mu \geq \mu_0, \quad w^T \mathbf{1} = 1, \quad w \geq 0$
ESG-constrained: Add $w^T s \geq s_0$ where s = ESG score vector, s_0 = minimum portfolio ESG score
Greenium $= \mu^*_{unconstrained} - \mu^*_{constrained}$ for the same risk level
Shadow price of ESG constraint: $\dfrac{\partial \sigma_p^2}{\partial s_0}$ = additional risk per unit of ESG improvement
Empirical evidence: Greenium is small (5–15 bps) for broad ESG tilts but large (50+ bps) for strict exclusions (fossil fuel free)
The paradox: ESG investing reduces the investable universe. In efficient markets, a smaller universe = same or worse risk-adjusted returns. But if ESG predicts risk (stranded assets, regulatory fines), ESG-aware portfolios may actually REDUCE tail risk.

The greenium is real but small for broad tilts. The economic case for ESG is not higher returns – it is lower tail risk from avoiding companies exposed to regulatory, environmental, or governance shocks.

How Does an ESG Tilt Move Your Portfolio on the Risk-Return Plane?



- Arrow/vector plot on risk-return plane with efficient frontier in gray
- Starting points: standard portfolios (Conservative, Moderate, Aggressive) shown as circles
- Three ESG strategies per portfolio:
  - Broad ESG tilt: short arrows, minimal impact
  - SRI exclusions: medium arrows, southwest shift
  - Impact investing: long arrows, meaningful sacrifice
- Broad tilts barely move you; strict exclusions push portfolios well inside the frontier

Broad ESG tilts barely move the needle – arrows are short. Strict exclusions push portfolios meaningfully inside the efficient frontier. The ESG strategy choice matters more than the ESG label.

Can You Build an ESG-Aware Robo-Advisor Allocation?

```python
import numpy as np
from scipy.optimize import minimize

def esg_robo(mu, cov, esg_scores, min_esg=65, rf=0.015):
    """Markowitz with ESG floor constraint."""
    n = len(mu)
    def neg_sharpe(w):
        ret = w @ mu
        vol = np.sqrt(w @ cov @ w)
        return -(ret - rf) / vol
    result = minimize(
        neg_sharpe, np.ones(n)/n, method='SLSQP',
        bounds=[(0, 1)] * n,
        constraints=[
            {'type': 'eq', 'fun': lambda w: np.sum(w) - 1},
            {'type': 'ineq', 'fun': lambda w: w @ esg_scores - min_esg}
        ])
    return result.x, -result.fun # weights, Sharpe
```

Adding one inequality constraint (ESG floor) to Markowitz is trivial. The hard part is getting reliable ESG scores – raters disagree 46% of the time (see L08).

Can a Robo-Advisor Discriminate Without Knowing Your Race?

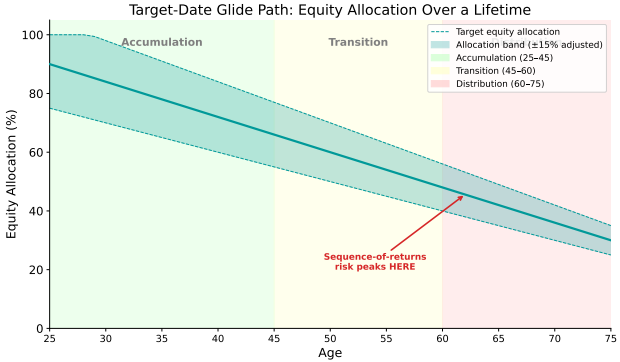
Robo-advisor inputs: Age, income, savings, risk questionnaire answers, employment type
Potential proxy discrimination:

- ZIP code correlates with race (redlining)
- Income level correlates with gender (pay gap)
- Employment type correlates with immigration status

Demographic parity: $P(\text{Aggressive profile} \mid A = 0) = P(\text{Aggressive profile} \mid A = 1)$
Equalized odds: $P(\text{Equity} > 60\% \mid \text{Tolerance} = H, A = 0) = P(\text{Equity} > 60\% \mid \text{Tolerance} = H, A = 1)$
The fiduciary paradox: If women genuinely have shorter investment horizons (career breaks, longer lifespan, lower average income), is recommending lower equity to women FAIR (matching risk to circumstances) or DISCRIMINATORY (perpetuating wealth gap)?
No mathematical resolution – this is a values question, identical to the impossibility theorem (Chouldechova 2017) from L03 credit scoring.

The fairness question in wealth management is identical to credit scoring: the algorithm reflects the data, and the data reflects historical inequality. No optimizer resolves a values question.

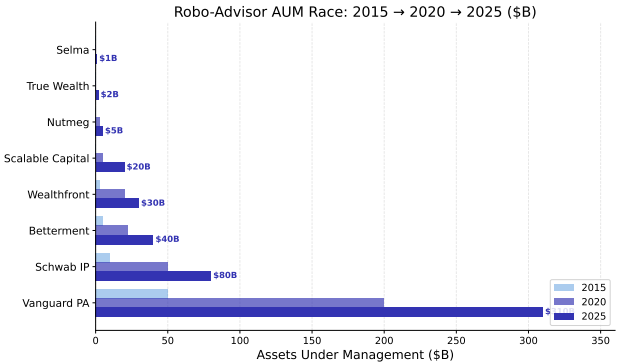
How Should Your Portfolio Evolve as You Approach Retirement?



- Area band chart: target-date fund glide path from age 25 to 75
- Central line = target equity allocation (declining from 90% to 30%)
- Upper/lower bands = acceptable range based on market conditions
- Three phases: **Accumulation** (25–45, wide band, high equity), **Transition** (45–60, narrowing band), **Distribution** (60–75, narrow band, low equity)
- "Sequence-of-returns risk is highest HERE" – ages 55–65, where a crash is catastrophic because there is no time to recover

Sequence-of-returns risk peaks in the 5 years before retirement. A 30% crash at age 63 is 5x more damaging than the same crash at age 35 because there is no time to recover.

Which Robo-Advisor Is Winning the AUM Race?



- Horizontal racing bar chart: AUM growth for 8 robo-advisors from 2015 to 2025
- Vanguard Personal Advisor dominates ($300B+), Schwab Intelligent is #2 ($80B)
- Pure-plays (Betterment $40B, Wealthfront $30B) are dwarfed by incumbents
- European entrants (Scalable Capital, Nutmeg) are visible but small
- Key insight: incumbents with existing client bases and distribution crush pure-play startups
- The "robo-advisor" is a feature, not a company

Vanguard's hybrid model ($300B+) dwarfs every pure-play robo-advisor combined. The lesson: robo-advisory is a FEATURE of wealth management, not a standalone business.

Can You Simulate How Retirement Timing Risk Destroys Wealth?

```python
import numpy as np

def glide_path_sim(start_age=25, retire=65, n_sims=5000):
    """Simulate wealth paths with declining equity glide."""
    yrs = retire - start_age
    np.random.seed(42)
    wealth = np.zeros((n_sims, yrs))
    for s in range(n_sims):
        w = 100_000
        for t in range(yrs):
            eq = max(0.3, 1.0 - (start_age + t - 25) * 0.015)
            r_eq = np.random.normal(0.07, 0.16)
            r_bd = np.random.normal(0.03, 0.04)
            w *= 1 + eq * r_eq + (1 - eq) * r_bd
            wealth[s, t] = w
    return wealth

print(f"Median at 65: ${np.median(glide_path_sim()[:,-1]):,.0f}")
```

Run 5,000 simulations to see the range of outcomes. The 5th percentile is what keeps financial planners up at night – sequence-of-returns risk is the retirement killer.

How Much Does the Fee Drag Actually Cost Over a Lifetime?

Compound wealth with fees: $W_T = W_0 \prod_{t=1}^{T}(1 + R_t - f)$
Expected terminal wealth: $E[W_T] = W_0(1 + \mu - f)^T$ (assuming i.i.d. returns)
Fee drag ratio:

$$\frac{W_T^{robo}}{W_T^{trad}} = \left(\frac{1 + \mu - f_{robo}}{1 + \mu - f_{trad}}\right)^T$$

Example: For $\mu = 7\%$, $f_{robo} = 0.25\%$, $f_{trad} = 1.50\%$, $T = 30$: ratio $= 1.42$ (robo investor has 42% more)
Breakeven complexity: At what portfolio size does a human advisor's tax optimization and estate planning outweigh the fee difference? $f_{trad} - f_{robo} = 1.25\%$ on W. Human advisor adds: $\sim$0.5% behavioral coaching + $\sim$0.3% tax optimization + $\sim$0.2% estate planning $= \sim$1.0%. Net cost of human: $\sim$0.25%.
Verdict: For most clients, the math favors robo or hybrid.

A human advisor must add at least 1.25% in annual value to justify the fee premium. Behavioral coaching (0.5%) plus tax optimization (0.3%) gets you close – but not quite there for most clients.

What Have We Learned – and What Remains Unsolved?

- **MPT Mathematics:** The efficient frontier is a parabola. The Lagrangian derivation is exact. But input estimation errors (especially expected returns) dominate output quality.
- **Black-Litterman:** Stabilizes Markowitz by anchoring to equilibrium. With no views, you get the market portfolio – the safest default. Risk parity goes further by eliminating return inputs entirely.
- **Rebalancing & TLH:** Optimal threshold balances transaction costs, tax drag, and risk drift. TLH adds 0.5–1.5% annually but is front-loaded. Both are where robo-advisors create measurable value.
- **Behavioral Finance:** Loss aversion ($\lambda = 2.25$) explains most investor irrationality. Risk questionnaires measure stated preferences, not revealed ones. Adjusting for bias shrinks recommended equity by $\sim$25%.
- **ESG Integration:** Broad ESG tilts cost 5–15 bps. Strict exclusions cost 50+ bps. The economic case is tail-risk reduction, not return enhancement.

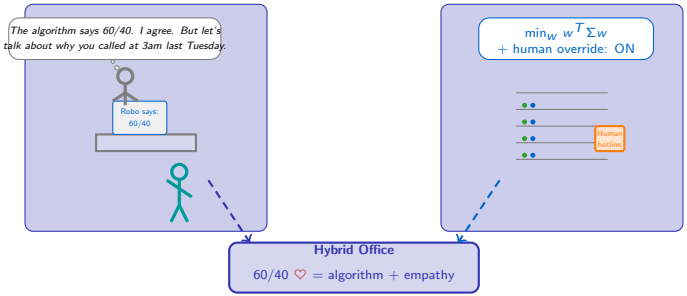
What remains unsolved: Sequence-of-returns risk in retirement, algorithmic fairness in wealth management, crisis-period model failures.

Five sections, one lesson: robo-advisors are elegant math wrapped in a behavioral challenge. The algorithm is right – but being right is not enough.

Key Takeaways

1. **MPT is the engine** – Lagrangian derivation gives exact efficient frontier. But "garbage in, garbage out": return estimation errors dominate output quality.
2. **Black-Litterman stabilizes Markowitz** by anchoring to market equilibrium. Risk parity goes further by eliminating return inputs entirely. Both are production alternatives to raw MPT.
3. **Rebalancing and TLH are where robo-advisors create measurable value** – estimated 0.5–1.5% annually in after-tax alpha. Optimal threshold follows the $\delta^* \propto (c/\sigma^2)^{1/3}$ rule.
4. **Behavioral biases cost 4–5% annually** (Dalbar). Loss aversion ($\lambda = 2.25$) is the biggest single factor. Robo-advisors automate away $\sim$60% of the behavioral gap.
5. **ESG integration is a constraint, not a free lunch** – broad tilts cost 5–15 bps, strict exclusions 50+ bps. The case for ESG is tail-risk reduction, not return enhancement.
6. **The market verdict is hybrid** – Vanguard Personal Advisor ($300B+) proves that algorithm + human access outcompetes pure robo or pure human. Robo-advisory is a feature, not a company.

Six takeaways, one framework: the math is solved. The behavioral challenge is not. The winning robo-advisor will be the one that bridges the gap between optimal portfolios and human psychology.



The best advisor is a machine that knows when to call a human.