

FinOpt-Workflow

December 29, 2025

1 FinOpt: Complete Workflow Validation

2 Configuration: Seasonality and Contribution

```
[2]: # --- Seasonality pattern for variable income (12 months, Jan-Dec) ---  
  
# Structure: months = ["Jan", "Feb", "Mar", "Apr", "May", "Jun",  
#                   "Jul", "Aug", "Sep", "Oct", "Nov", "Dec"]  
  
seasonality_variable = [0.00, 0.00, 0.00, 0.6, 1, 1.16,  
                        1, 1.10, 0.50, 0.90, 0.85, 1]  
  
monthly_contrib_fixed = [0.45, 0.45, 0.45, 0.45, 0.45, 0.45,  
                        0.45, 0.45, 0.45, 0.45, 0.45, 0.3]  
  
monthly_contrib_variable = [1.0] * 12  
  
monthly_contribution = {"fixed": monthly_contrib_fixed, "variable":  
    ↪monthly_contrib_variable}
```

3 Initialize FinancialModel

```
[3]: # IncomeModel Instantiation  
income = IncomeModel(  
    fixed=FixedIncome(  
        base=1_488_000.0,      # CLP/month  
        annual_growth=0.03,    # 3% nominal annual growth  
        salary_raises={  
            date(2026, 4, 1): 400_000, # +400k in April 2026  
            date(2027, 6, 1): 400_000, # +400k in Jun 2027  
            date(2028, 6, 1): 300_000  # +300k in Jun 2028  
        },  
        name="fixed"  
    ),  
    variable=VariableIncome(  
        base=60_000.0,        # Base variable income  
        seasonality=seasonality_variable,
```

```

        sigma=0.10,          # 10% monthly noise
        floor=0.0,          # No negative income
        cap=400_000.0,      # Maximum 400k/month
        annual_growth=0.0,  # No growth in variable
        name="variable"
    ),
    monthly_contribution = monthly_contribution
)

# --- Account configuration ---
accounts = [
    Account.from_annual(
        name="Cuenta Ahorro Vivienda (BE)",
        annual_return=0.025,
        annual_volatility=0.01,
        initial_wealth=1600000
    ),
    Account.from_annual(
        name="Conservative Clooney (Fintual)",
        annual_return=0.08,
        annual_volatility=0.09,
        initial_wealth=744747
    ),
    Account.from_annual(
        name="Risky Norris (Fintual)",
        annual_return=0.14,
        annual_volatility=0.15,
        initial_wealth=900000
    )
]

# --- Correlation matrix (2x2) ---
# UF portfolio have moderate positive correlation (rho = 0.2)
correlation_matrix = np.array([
    [1.0, 0.0, 0.0],
    [0.0, 1.0, 0.5],
    [0.0, 0.5, 1.0]
])

# --- Initialize FinancialModel ---
model = FinancialModel(income, accounts, default_correlation = 
    correlation_matrix)
model

```

[3]: FinancialModel(M=3, accounts=['Cuenta Ahorro Vivienda (BE)', 'Conservative Clooney (Fintual)', 'Risky Norris (Fintual)'], cache=enabled)

4 Simulation parameters

```
[4]: # --- Simulation parameters ---  
n_sims = 500  
months = 26  
start_date = date(2025, 11, 1)
```

5 Income Module

Total monthly income at time t is composed of fixed and variable parts:

$$Y_t = y_t^{\text{fixed}} + Y_t^{\text{variable}}$$

5.1 Fixed Income

The fixed component, y_t^{fixed} , reflects a baseline salary subject to compounded annual growth g and scheduled raises d_k, Δ_k (e.g., promotions or tenure milestones):

$$y_t^{\text{fixed}} = \text{current_salary}(t) \cdot (1 + m)^{\Delta t}$$

where $m = (1 + g)^{1/12} - 1$ is the **monthly compounded rate**, and Δt represents time since the last raise.

5.2 Variable Income

The variable component, Y_t^{variable} , models irregular income sources (e.g., freelance work or bonuses) with:

- **Seasonality:** $s \in \mathbb{R}^{12}$ (multiplicative monthly factors),
- **Noise:** $\epsilon_t \sim \mathcal{N}(0, \sigma^2)$ (Gaussian shocks),
- **Growth:** same compounded rate m applied to a base income level,
- **Boundaries:** optional floor and cap constraints.

The underlying stochastic projection is:

$$\tilde{Y}_t = \max(\text{floor}, \mu_t(1 + \epsilon_t)), \quad \text{where } \mu_t = \text{base} \cdot (1 + m)^t \cdot s_{(t \bmod 12)}$$

Then, guardrails are applied as:

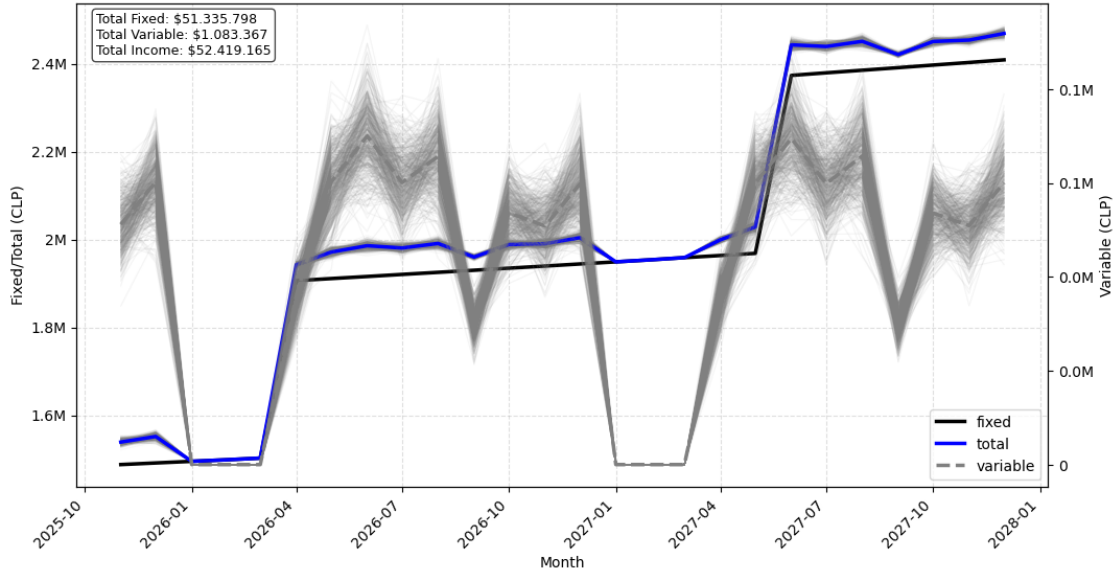
$$Y_t^{\text{variable}} = \begin{cases} 0 & \text{if } \tilde{Y}_t < 0 \\ \tilde{Y}_t & \text{if } 0 \leq \tilde{Y}_t \leq \text{cap} \\ \text{cap} & \text{if } \tilde{Y}_t > \text{cap} \end{cases}$$

Note: In expectation (ignoring noise truncation), $\mathbb{E}[Y_t] = y_t^{\text{fixed}} + \mu_t$

5.3 Income Projection

Dual-axis plot with: - **Left axis:** Fixed income (deterministic) + Total income - **Right axis:** Variable income (stochastic with trajectories) - **Trajectories:** Individual noise realizations (n=300) - **Confidence band:** 95% CI for variable income

```
[5]: # Income Projection Simulation
model.plot(mode='income', T=months, start=start_date)
```



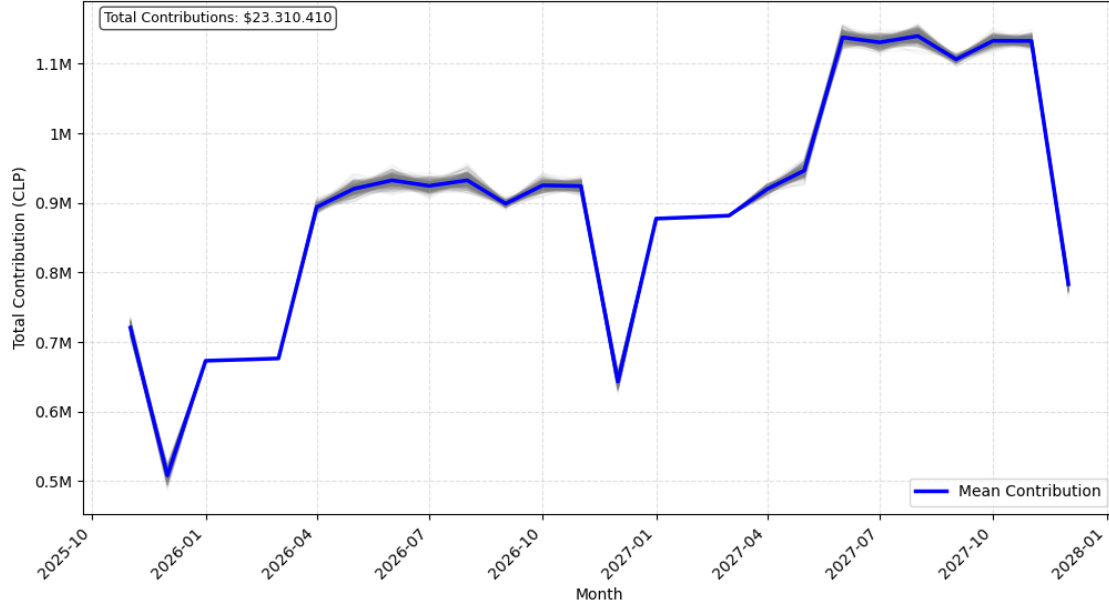
5.4 Contribution Projection

A fraction of income is allocated each month through calendar-rotating schedules:

$$A_t = \alpha_{(t \bmod 12)}^f \cdot y_t^{\text{fixed}} + \alpha_{(t \bmod 12)}^v \cdot Y_t^{\text{variable}}$$

where $\alpha^f, \alpha^v \in [0, 1]^{12}$ control the fixed and variable contribution rates by applying the 12-month fractional arrays to projected incomes, rotated according to **start** date and repeated cyclically for horizons > 12 months.

```
[6]: # Contribution Projection Simulation
model.plot(mode='contributions', T=months, start=start_date)
```



6 Return Module

6.1 Multi-Account Return Model

For M accounts with correlated returns:

$$1 + R_t^m \sim \text{LogNormal}(\mu_{\log}^m, \Sigma)$$

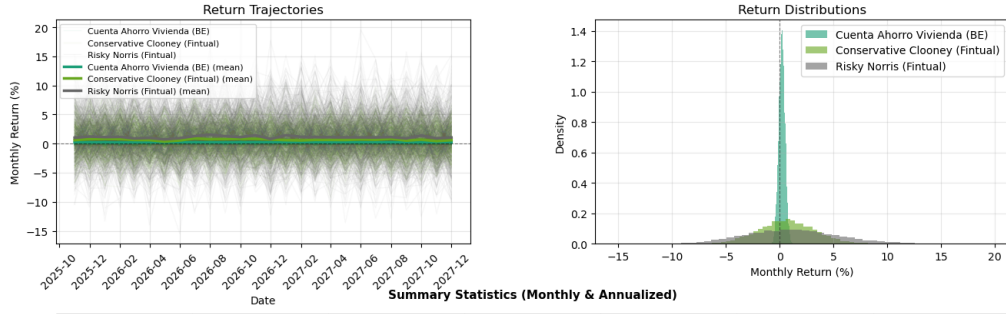
where $\Sigma = D \cdot \rho \cdot D$ is the covariance matrix: - $D = \text{diag}(\sigma_{\log}^1, \dots, \sigma_{\log}^M)$ - $\rho \in \mathbb{R}^{M \times M}$ is the correlation matrix (symmetric, PSD)

6.2 Monthly Return Distribution (Marginal Analysis)

Visualizes IID monthly returns across both accounts with 4 panels: - **Trajectories**: Individual paths for each account - **Histograms**: Marginal distributions (overlaid) - **Statistics**: Mean, std, quantiles per account

```
[7]: model.plot(mode = 'returns', T = months, start=start_date)
```

Monthly Return Distributions (R_t^m)



Summary Statistics (Monthly & Annualized)

Account	Mean	Std	Median	IQR (Q25-Q75)	Min	Max
Cuenta Ahorro Vivienda (BE)	0.20%	0.29%	0.20%	[0.01%, 0.40%]	-0.97%	1.55%
↳ Annualized	2.48%	1.00%	2.46%	[0.07%, 4.89%]	—	—
Conservative Clooney (Fintual)	0.66%	2.60%	0.67%	[-1.08%, 2.44%]	-9.29%	12.04%
↳ Annualized	8.27%	9.02%	8.33%	[-12.25%, 33.47%]	—	—
Risky Norris (Fintual)	1.11%	4.37%	1.00%	[-1.89%, 3.95%]	-15.46%	19.57%
↳ Annualized	14.17%	15.15%	12.70%	[-20.50%, 59.20%]	—	—

T=26 months | n_sims=500 | seed=None

6.3 Cumulative Returns per Account

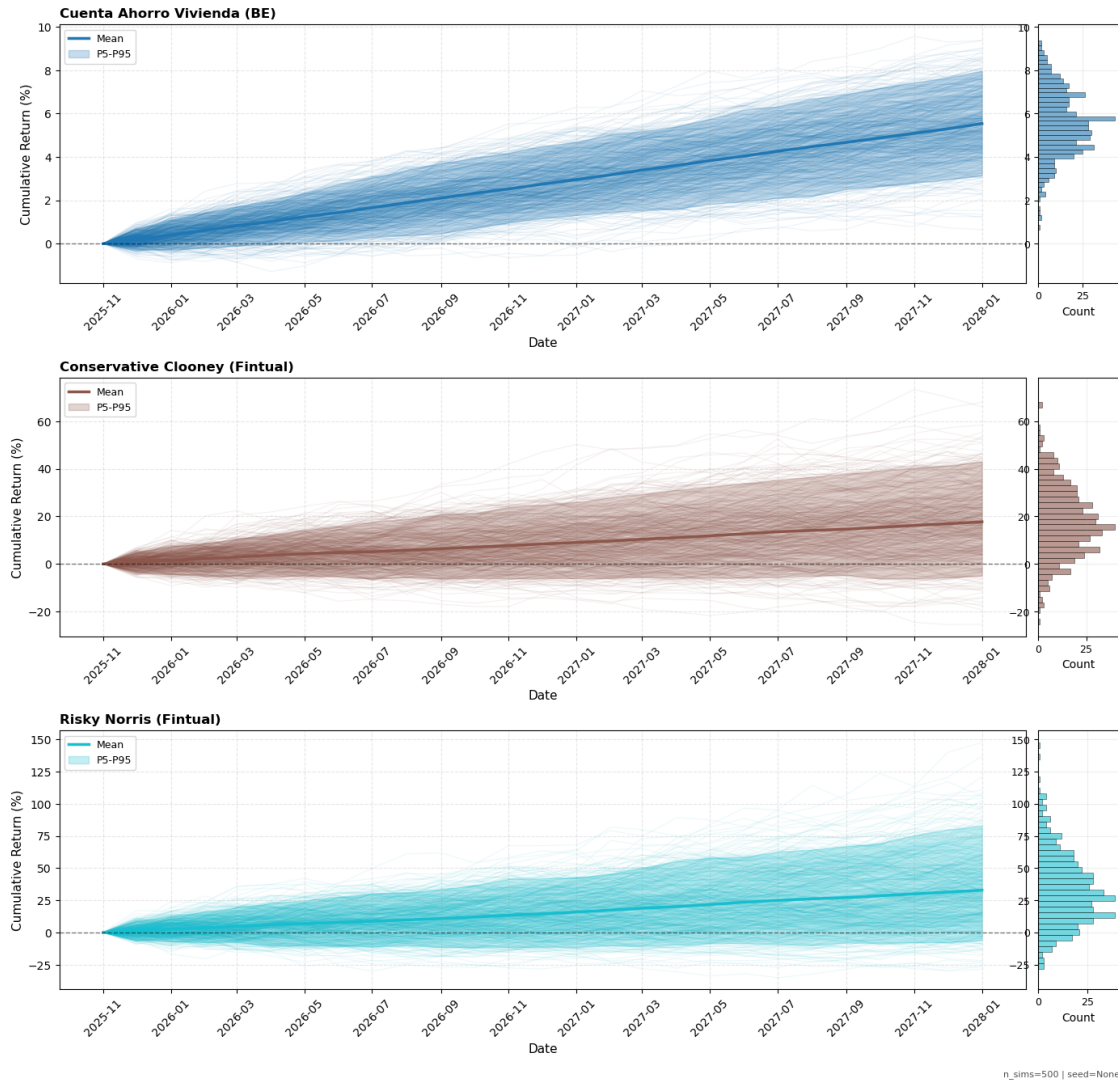
For M accounts with correlated returns:

$$R_{cm}^m(T) = \prod_{t=0}^{T-1} (1 + R_t^m) - 1$$

Cross-sectional correlation persists through time but does not compound.

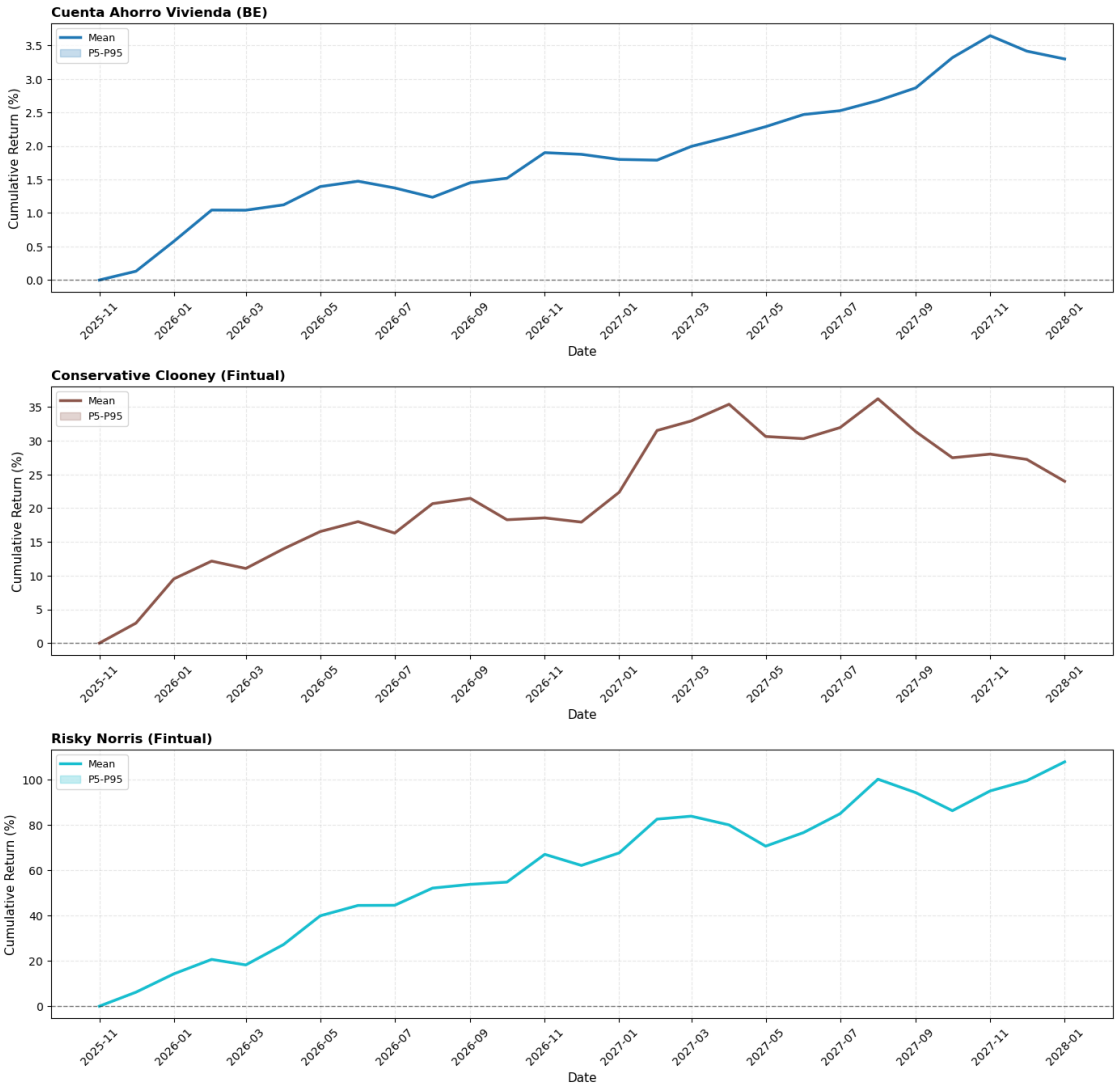
```
[8]: model.plot(mode = 'returns_cumulative', T = months, start=start_date)
```

Cumulative Returns per Account $\left(\prod_{t=0}^{T-1} (1 + R_t^m) - 1 \right)$



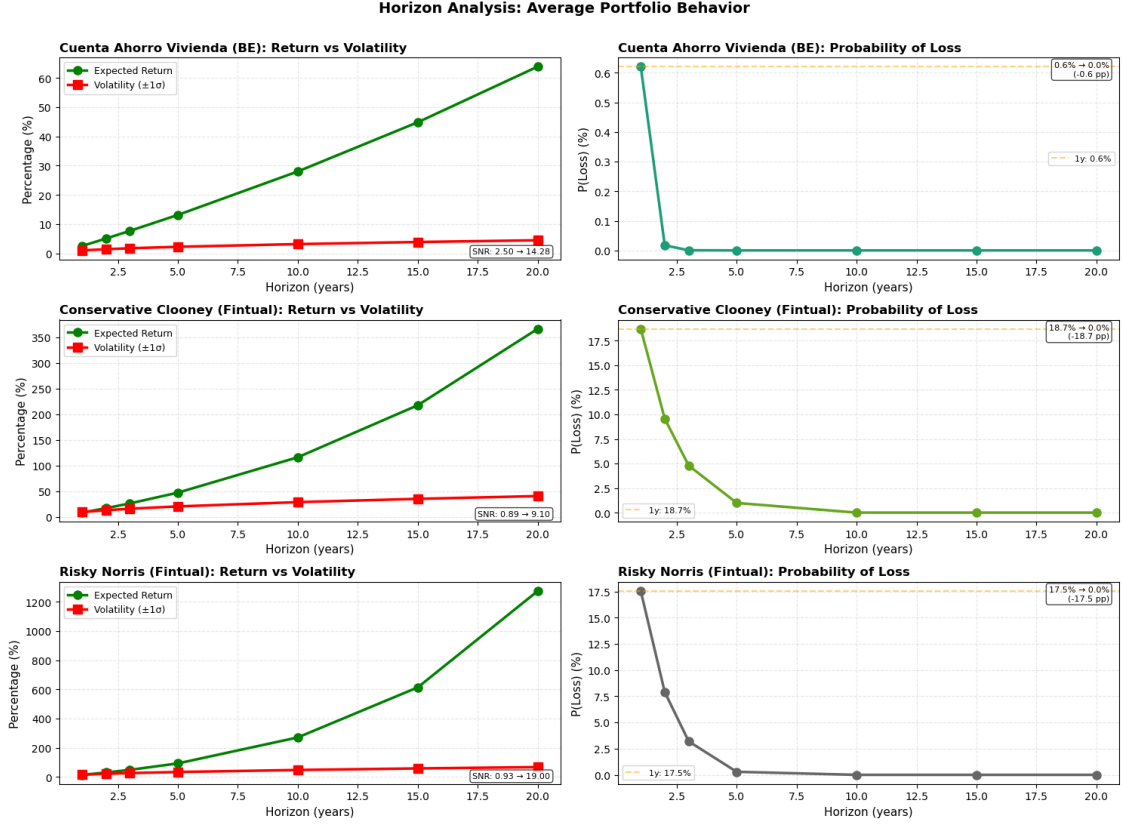
```
[9]: model.plot(mode = 'returns_cumulative', T = months, start=start_date, n_sims=1,
    ↪title= r'Random Example of Cumulative Returns per Account'
    ↪$(\prod_{t=0}^{T-1}(1 + R_t^m) - 1)')
```

Random Example of Cumulative Returns per Account $\prod_{t=0}^{T-1} (1 + R_t^m) - 1$



n_sims=1 | seed=None

6.4 Horizon Analysis: Time Diversification by Account



7 Portfolio Module

7.1 Wealth Projection Under Allocation Policy

Recursive dynamics: Wealth evolves as:

$$W_{t+1}^m = (W_t^m + A_t x_t^m) (1 + R_t^m)$$

We define $A_t^m = A_t \cdot x_t^m$ which is the contribution allocated to account m via policy $X = \{x_t^m\}_{t,m}$.

Closed-form representation:

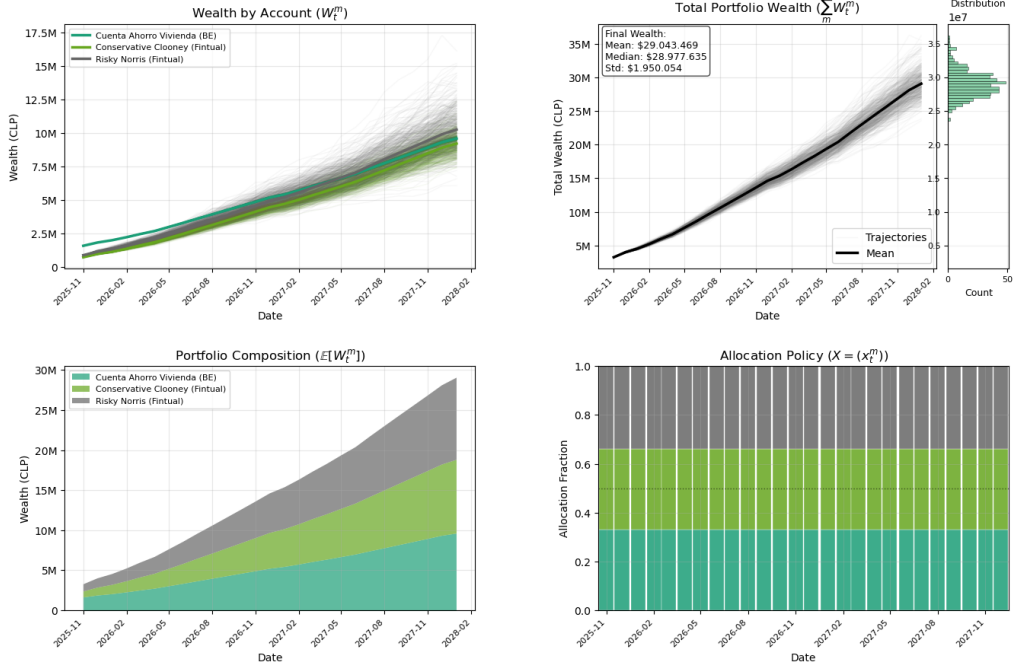
$$W_t^m = W_0^m F_{0,t}^m + \sum_{s=0}^{t-1} A_s x_s^m F_{s,t}^m$$

with accumulation factor $F_{s,t}^m = \prod_{r=s}^{t-1} (1 + R_r^m)$.

Key insight: $W_t^m(X)$ is linear affine in policy $X \rightarrow$ analytical gradients enable convex optimization.

What to observe: - Top-left: wealth per account with Monte Carlo trajectories - Top-right: total wealth + final distribution histogram - Bottom-left: portfolio composition over time - Bottom-right: allocation policy heatmap

/home/mlloi/fin-opt/src/portfolio.py:1148: UserWarning: This figure includes Axes that are not compatible with tight_layout, so results might be incorrect.
fig.tight_layout(rect=[0, 0.01, 1, 0.96 if title else 1])



8 Goal-Driven Optimization

8.1 Problem Formulation

8.1.1 Non-Convex Chance Constraints

The **bilevel problem** seeks the minimum horizon T^* satisfying all goals:

$$\min_{T \in \mathbb{N}} T \max_{X \in \mathcal{F}_T} f(X)$$

where the **goal-feasible set** \mathcal{F}_T contains all policies $X \in \mathcal{X}_T$ satisfying probabilistic constraints:

Intermediate goals (at fixed time $t < T$):

$$\mathbb{P}(W_t^m(X) \geq b_t^m) \geq 1 - \varepsilon_t^m, \quad \forall g \in \mathcal{G}_{\text{int}}$$

Terminal goals (at horizon T):

$$\mathbb{P}(W_T^m(X) \geq b^m) \geq 1 - \varepsilon^m, \quad \forall g \in \mathcal{G}_{\text{term}}$$

with decision space (simplex):

$$\mathcal{X}_T = \left\{ X \in \mathbb{R}^{T \times M} : x_t^m \geq 0, \sum_{m=1}^M x_t^m = 1, \forall t = 0, \dots, T-1 \right\}$$

Challenge: Chance constraints involve indicator functions $\mathbb{1}[\cdot]$, which are discontinuous and non-convex. Standard approaches (MILP, sigmoid smoothing) either scale poorly or find local optima.

8.2 Convex Reformulation via CVaR

8.2.1 CVaR Reformulation: Convex Upper Bound

We replace each chance constraint with a **CVaR constraint** (Rockafellar & Uryasev, 2000):

$$\boxed{\mathbb{P}(W \geq b) \geq 1 - \varepsilon \quad \Longleftarrow \quad \text{CVaR}_\varepsilon(b - W) \leq 0}$$

where the **Conditional Value-at-Risk** of shortfall $L = b - W$ is:

$$\text{CVaR}_\varepsilon(L) = \text{VaR}_\varepsilon(L) + \frac{1}{\varepsilon} \mathbb{E}[(L - \text{VaR}_\varepsilon(L))_+]$$

Epigraphic formulation (convex, suitable for LP solvers):

$$\text{CVaR}_\varepsilon(L) = \min_{\gamma \in \mathbb{R}} \left\{ \gamma + \frac{1}{\varepsilon N} \sum_{i=1}^N [L^i - \gamma]_+ \right\}$$

Introducing auxiliary variables $z^i \geq [L^i - \gamma]_+$:

$$\begin{aligned} \text{CVaR}_\varepsilon(L) = \min_{\gamma, z} & \left\{ \gamma + \frac{1}{\varepsilon N} \sum_{i=1}^N z^i \right\} \\ \text{s.t.} \quad & z^i \geq L^i - \gamma, \quad \forall i \in [N] \\ & z^i \geq 0, \quad \forall i \in [N] \end{aligned}$$

Key property: If W^i is **affine in X** (as in our wealth dynamics), then $\text{CVaR}_\varepsilon(b - W)$ is **convex in X** .

8.2.2 Mathematical Relationship: Implication, Not Equivalence

Theorem (Rockafellar & Uryasev, 2000):

$$\text{CVaR}_\varepsilon(L) \leq 0 \quad \implies \quad \mathbb{P}(L \leq 0) \geq 1 - \varepsilon$$

Proof sketch: CVaR averages the worst ε -tail of the distribution. If this mean is non-positive, then at least $(1 - \varepsilon)$ of scenarios satisfy $L \leq 0$.

The converse is NOT true: $\mathbb{P}(L \leq 0) \geq 1 - \varepsilon$ does NOT imply $\text{CVaR} \leq 0$ if the tail is heavy.

Interpretation: CVaR is a **conservative approximation** controlling both: 1. **Frequency** of violations (at most $\varepsilon \times 100\%$ scenarios fail) 2. **Severity** of violations (average loss in tail is non-positive)

The original chance constraint only controls frequency.

8.2.3 Convex Reformulated Problem - Multiple Objectives

Inner problem (for fixed T):

$$\begin{aligned}
& \max_{X, \gamma, z} f(X) \\
& \text{s.t.} \quad \sum_{m=1}^M x_t^m = 1, \quad \forall t = 0, \dots, T-1 \quad (\text{simplex}) \\
& \quad x_t^m \geq 0, \quad \forall t, m \quad (\text{non-negativity}) \\
& \quad z_g^i \geq (b_g - W_{t_g}^{i, m_g}(X)) - \gamma_g, \quad \forall g, i \quad (\text{epigraph}) \\
& \quad z_g^i \geq 0, \quad \forall g, i \\
& \quad \gamma_g + \frac{1}{\varepsilon_g N} \sum_{i=1}^N z_g^i \leq 0, \quad \forall g \quad (\text{CVaR constraint})
\end{aligned}$$

where: - $W_t^{i, m}(X) = W_0^m F_{0, t, m}^i + \sum_{s=0}^{t-1} A_s^i \cdot x_s^m \cdot F_{s, t, m}^i$ (**affine in X**) - g indexes goals (both intermediate and terminal) - $t_g, m_g, b_g, \varepsilon_g$ are parameters of goal g - $f(X)$ is a convex objective function (see supported objectives below)

Global optimality guaranteed via convex programming (interior-point methods).

8.2.4 Comparison: Original vs. CVaR

Observed conservativeness (empirical): - CVaR constraint: $\text{CVaR}_\varepsilon(L) \leq 0$ - Resulting violation rate: typically $(0.5 - 0.8) \times \varepsilon$ (better than required)

Example (from our results): - Goal: $\mathbb{P}(W_T \geq 1M) \geq 80\%$ (i.e., $\varepsilon = 20\%$) - CVaR solution: violation rate = 9% (margin of 11%)

The 11% buffer is the “price” of convexity, buying us certified global optimality and numerical stability.

8.3 Goal Specification

FINANCIAL GOALS SUMMARY

1.

Type: Terminal (horizon T to be optimized)
Account: Conservative Clooney (Fintual)
Threshold: \$5,000,000
Minimum Confidence: 50% (=50%)

2.

Type: Terminal (horizon T to be optimized)
Account: Risky Norris (Fintual)
Threshold: \$8,000,000
Minimum Confidence: 50% (=50%)

3.

Type: Intermediate (month 12)
Account: Cuenta Ahorro Vivienda (BE)
Threshold: \$3,100,000
Minimum Confidence: 50% (=50%)

4.

Type: Intermediate (month 18)
Account: Conservative Clooney (Fintual)
Threshold: \$3,000,000
Minimum Confidence: 50% (=50%)

=====

8.4 Bilevel Optimization

8.4.1 Supported Convex Objectives

CVaROptimizer supports 7 convex objectives exploiting affine wealth structure: Consider $W_t^{i,m}(X) = W_0^m F_{0,t,m}^i + \sum_{s=0}^{t-1} A_s^i \cdot x_s^m \cdot F_{s,t,m}^i$ and $\mathbb{E} \left[\sum_{m=1}^M W_T^m \right] = \frac{1}{N} \sum_{i=1}^N \sum_{m=1}^M W_T^{i,m}(X)$.

1. **risky** — Growth maximization

$$\max_X \quad \mathbb{E} \left[\sum_{m=1}^M W_T^m \right]$$

2. **balanced** — Equal contribution across time

$$\max_X \quad - \sum_{t=1}^{T-1} \sum_{m=1}^M (x_t^m - x_{t-1}^m)^2$$

3. **risky_turnover** — Growth maximization with penalty over contributions across time

$$\max_X \quad \mathbb{E} \left[\sum_{m=1}^M W_T^m \right] - \sum_{t=1}^{T-1} \sum_{m=1}^M (x_t^m - x_{t-1}^m)^2$$

4. **conservative** — Minimize variance, chose conservative portfolio

$$\max_X \quad \mathbb{E} \left[\sum_{m=1}^M W_T^m \right] - \lambda \cdot \text{Var} \left(\sum_{m=1}^M W_T^m \right)$$

```
[13]: # --- Execute Optimization ---
```

```
print("\n" + "=" * 70)
print("STARTING BILEVEL OPTIMIZATION")

objective_params = {'lambda' : 50}
optimizer = CVaROptimizer(n_accounts=model.M, objective='balanced',
    ↪objective_params=objective_params)
opt_result = model.optimize(
    goals=goals,
    optimizer=optimizer,
    T_max=120,
    n_sims=400,
    seed=42,
    verbose=True,
    solver='ECOS', # 0 'SCS', 'CLARABEL'
    max_iters=10000)

print(opt_result.summary())

# Display optimization summary
print("\n" + "=" * 70)
print(opt_result.summary())
print("=" * 70)
```

```
=====
STARTING BILEVEL OPTIMIZATION
```

```
=== GoalSeeker: BINARY search T [18, 120] ===
[Iter 1] Binary search: testing T=69 (range=[18, 120])...
/home/mlloi/anaconda3/envs/finance/lib/python3.11/site-
packages/cvxpy/problems/problem.py:1539: UserWarning: Solution may be
inaccurate. Try another solver, adjusting the solver settings, or solve with
verbose=True for more information.
  warnings.warn(
    Feasible, obj=-0.00, time=0.776s

[Iter 2] Binary search: testing T=43 (range=[18, 69])...
    Feasible, obj=-0.00, time=0.257s

[Iter 3] Binary search: testing T=30 (range=[18, 43])...
    Feasible, obj=-0.00, time=0.120s

[Iter 4] Binary search: testing T=24 (range=[18, 30])...
    Feasible, obj=-0.00, time=0.083s
```

```
[Iter 5] Binary search: testing T=21 (range=[18, 24])...  
Feasible, obj=-0.00, time=0.075s
```

```
[Iter 6] Binary search: testing T=19 (range=[18, 21])...  
Feasible, obj=-0.03, time=0.162s
```

```
[Iter 7] Binary search: testing T=18 (range=[18, 19])...  
Feasible, obj=-0.74, time=8.286s
```

```
=== Optimal: T*=18 (binary search converged) ===
```

```
OptimizationResult(  
    Status: Feasible  
    Horizon: T=18 months  
    Objective: -0.74  
    Goals: 4 (2 intermediate, 2 terminal)  
    Solve time: 8.286s  
    Iterations: 0  
)
```

```
=====  
OptimizationResult(  
    Status: Feasible  
    Horizon: T=18 months  
    Objective: -0.74  
    Goals: 4 (2 intermediate, 2 terminal)  
    Solve time: 8.286s  
    Iterations: 0  
)  
=====
```

```
[14]: optimizer.objective
```

```
[14]: 'balanced'
```

8.5 Optimal Policy Analysis

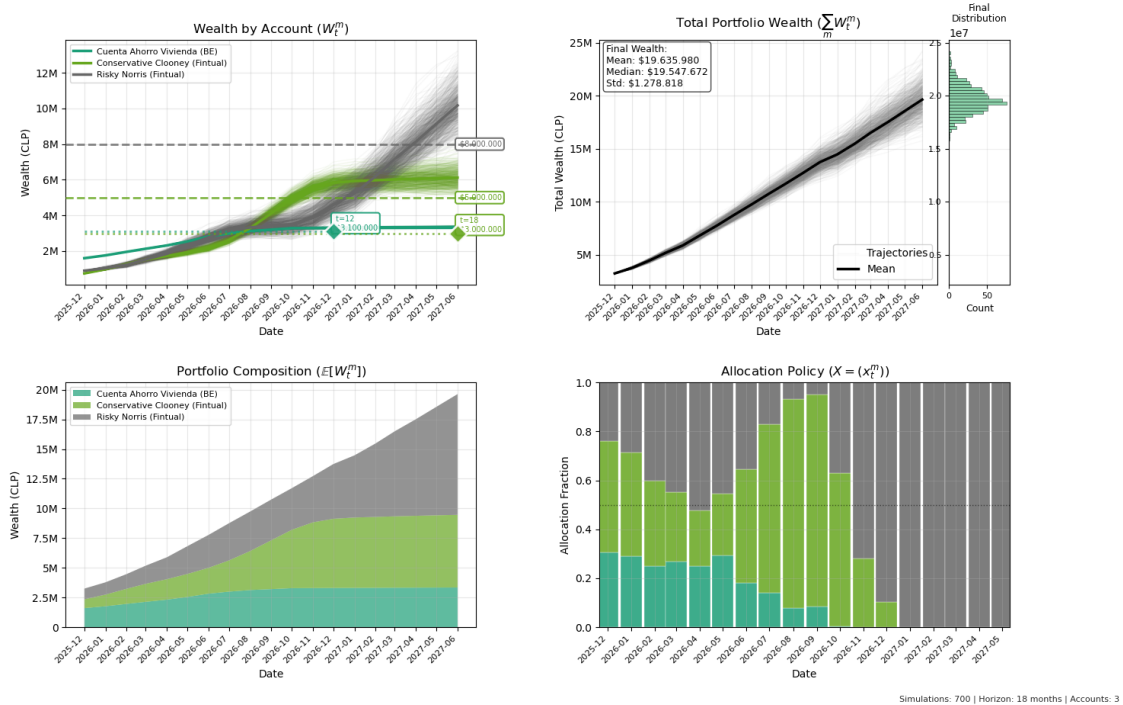
```
[15]: # --- Simulate wealth under X* ---  
  
# Generate fresh scenarios for validation (out-of-sample)  
opt_sim = model.simulate_from_optimization(  
    opt_result,  
    n_sims=700,  
    seed=999 # Different seed from optimization  
)  
  
# Plot wealth dynamics
```

```

model.plot(
    mode='wealth',
    result=opt_sim,
    X=opt_result.X,
    title=f"Wealth Dynamics under Optimal Policy (T*={opt_result.T})",
    show_trajectories=True,
    goals=goals
)

```

Wealth Dynamics under Optimal Policy (T*=18)



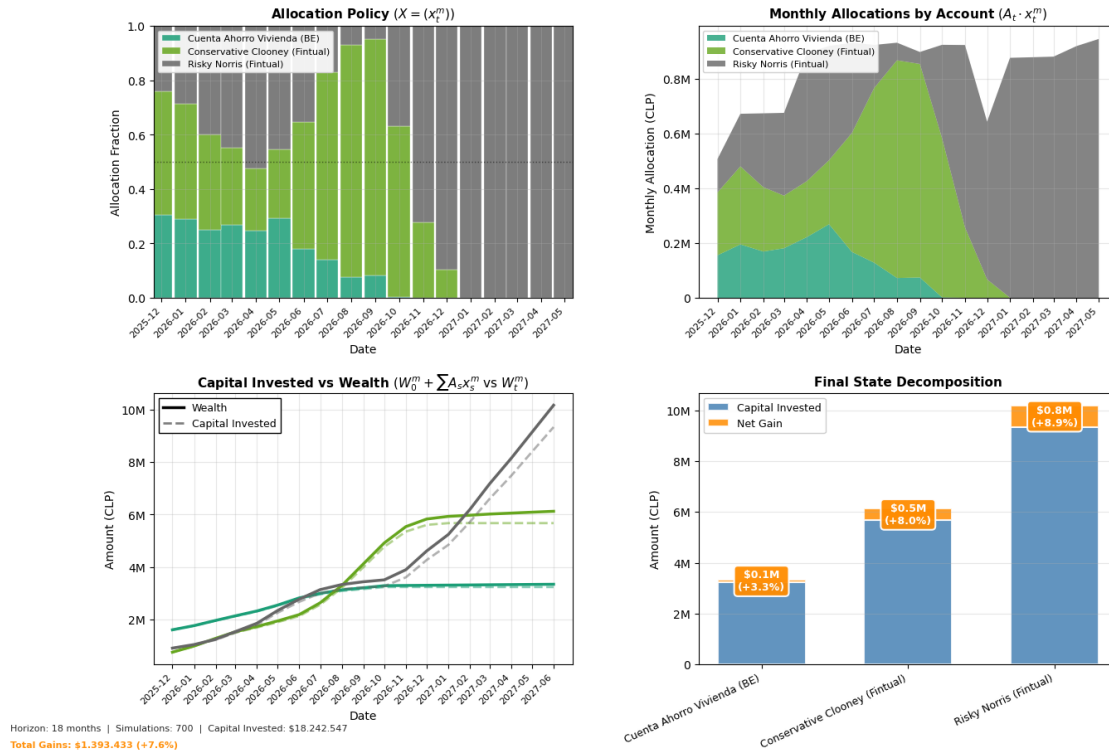
```

[16]: # Plot allocation analysis with investment gains
model.plot(
    mode='allocation',
    result=opt_sim,
    X=opt_result.X,
    title=f"Allocation Analysis under Optimal Policy (T*={opt_result.T})",
    show_trajectories=False
)

```

/home/mlloi/fin-opt/src/model.py:1895: UserWarning: This figure includes Axes that are not compatible with tight_layout, so results might be incorrect.
fig.tight_layout(rect=[0, 0.05, 1, 0.97 if title else 0.99])

Allocation Analysis under Optimal Policy (T*=18)



8.6 Goal Verification

8.6.1 In-Sample Verification

=====

GOAL VERIFICATION: IN-SAMPLE

=====

Using optimization scenarios (n=300, seed=42)

Expected: All goals satisfied ()

Account: Conservative Clooney (Fintual)
 Threshold: \$5,000,000
 Minimum Confidence: 50%

 Status: SATISFIED
 Violation rate: 1.00% (required 50.00%)
 Margin: 49.00%
 Median shortfall: \$94,448
 N violations: 3 / 300
 Account: Risky Norris (Fintual)
 Threshold: \$8,000,000
 Minimum Confidence: 50%

Status: SATISFIED
Violation rate: 1.33% (required 50.00%)
Margin: 48.67%
Median shortfall: \$341,149
N violations: 4 / 300
Account: Cuenta Ahorro Vivienda (BE)
Month: 12
Threshold: \$3,100,000
Minimum Confidence: 50%

Status: SATISFIED
Violation rate: 0.00% (required 50.00%)
Margin: 50.00%
Median shortfall: \$0
N violations: 0 / 300
Account: Conservative Clooney (Fintual)
Month: 18
Threshold: \$3,000,000
Minimum Confidence: 50%

Status: SATISFIED
Violation rate: 0.00% (required 50.00%)
Margin: 50.00%
Median shortfall: \$0
N violations: 0 / 300

=====

8.6.2 Out-of-Sample Verification

=====

GOAL VERIFICATION: OUT-OF-SAMPLE

=====

Using fresh scenarios (n=1000, seed=999)
Expected: All goals satisfied with safety margin

Account: Conservative Clooney (Fintual)
Threshold: \$5,000,000
Minimum Confidence: 50%

Status: SATISFIED
Violation rate: 0.50% (required 50.00%)
Margin: 49.50%
Median shortfall: \$92,858
N violations: 5 / 1000
Account: Risky Norris (Fintual)

Threshold: \$8,000,000
 Minimum Confidence: 50%

 Status: SATISFIED
 Violation rate: 0.50% (required 50.00%)
 Margin: 49.50%
 Median shortfall: \$404,092
 N violations: 5 / 1000
 Account: Cuenta Ahorro Vivienda (BE)
 Month: 12
 Threshold: \$3,100,000
 Minimum Confidence: 50%

 Status: SATISFIED
 Violation rate: 0.00% (required 50.00%)
 Margin: 50.00%
 Median shortfall: \$0
 N violations: 0 / 1000
 Account: Conservative Clooney (Fintual)
 Month: 18
 Threshold: \$3,000,000
 Minimum Confidence: 50%

 Status: SATISFIED
 Violation rate: 0.00% (required 50.00%)
 Margin: 50.00%
 Median shortfall: \$0
 N violations: 0 / 1000

=====

=====

IN-SAMPLE vs OUT-OF-SAMPLE COMPARISON

=====

	In-Sample (n=300)	Out-of-Sample (n=1000)	Required ()
0	0.0100	0.005	0.5
1	0.0133	0.005	0.5
2	0.0000	0.000	0.5
3	0.0000	0.000	0.5

=====

[NbConvertApp] Converting notebook FinOpt-Workflow.ipynb to pdf
 [NbConvertApp] Support files will be in FinOpt-Workflow_files/
 [NbConvertApp] Making directory ./FinOpt-Workflow_files
 [NbConvertApp] Writing 54000 bytes to notebook.tex
 [NbConvertApp] Building PDF
 [NbConvertApp] Running xelatex 3 times: ['xelatex', 'notebook.tex', '-quiet']
 [NbConvertApp] Running bibtex 1 time: ['bibtex', 'notebook']

```
[NbConvertApp] WARNING | bibtex had problems, most likely because there were no
citations
[NbConvertApp] PDF successfully created
[NbConvertApp] Writing 1859487 bytes to FinOpt-Workflow.pdf
[NbConvertApp] Converting notebook FinOpt-Workflow.ipynb to html
[NbConvertApp] WARNING | Alternative text is missing on 7 image(s).
[NbConvertApp] Writing 3059217 bytes to FinOpt-Workflow.html
```

[]:

