Market Simulator/Back tester – Ethan Alan John

What I set out to do

The goal of this project was to **build a realistic market simulator** that could model how a diversified portfolio might behave under different market conditions.
Instead of relying on historical averages, I wanted to **replicate market behavior** — how returns fluctuate, how assets move together, and how risk evolves over time.

In simple terms, the simulator tries to **"recreate the market"** using patterns learned from data — like how volatility rises in crises or how certain stocks tend to move together — and then **run thousands of alternate market paths** to see the possible outcomes for a portfolio.

Data & setup

I used **daily price data** from a broad set of stocks over several years.
The data was cleaned, aligned by date, and converted into daily percentage returns.

To keep things realistic and unbiased:

- All stocks were given **equal weight** (no stock-picking advantage).
- Any missing data was smoothly filled to avoid gaps.
- The model ran over roughly a **one-year horizon** across hundreds of simulated paths — enough to see both ordinary and extreme scenarios.

Method overview

The simulator is made up of **five interconnected parts**, each addressing a different layer of market behaviour.

1. Factor model — capturing market structure

I started by identifying the **common forces** that drive multiple stocks together.
Using historical returns, I built a **factor model** that grouped stocks into pseudo-sectors based on how correlated their returns were.
Each stock's behavior was then broken down into:

- A **market factor** (overall market movement),
- A **sector factor** (how similar companies move together), and
- A **style factor** (momentum based on the past 30 days).

2. GARCH engine — modeling volatility cycles

Markets don't move with constant volatility. They go through **quiet phases and sudden bursts of turbulence**.
To replicate that, I used a **GARCH (Generalized AutoRegressive Conditional Heteroskedasticity)** model for each factor.
This allowed volatility to evolve over time rather than stay fixed — so simulated markets could go from calm to chaotic just like in reality.

Each stock's "beta" — or sensitivity — to these factors was estimated using regression.
This step ensured that the simulator understood both broad market trends and sector-specific differences.

3. Copula engine — linking individual stocks

Even after accounting for common factors, stocks still move together in complex ways — sometimes tightly, sometimes loosely.
To capture this, I used a **Gaussian Copula**, which models how residuals (the parts not explained by factors) move together.
It ensures that when markets are under stress, correlations can strengthen, reflecting how real crises often cause everything to fall together.

4. Regime model — identifying market phases

I added a **Regime Model** that classifies each day into different market environments — typically "calm" or "volatile."
This was done using an **EWMA (Exponentially Weighted Moving Average)** of overall volatility, marking periods that cross a certain threshold as "high-risk regimes."
This makes the simulator adaptive: it behaves differently depending on whether the market is stable or under stress.

5. Portfolio simulator and metrics — bringing it all together

Finally, the **Portfolio Simulator** combines all these elements:

- It generates factor movements from the GARCH model,
- Draws residuals through the Copula,

- Adjusts each stock’s risk profile based on its regime and betas, and
- Aggregates them into portfolio returns.

For each simulated path, I measured:

- **Average return and volatility**,
- **Sharpe and Sortino ratios** (risk-adjusted performance),
- **Maximum drawdown** (worst peak-to-trough fall), and
- **Tail risk metrics** such as **Value-at-Risk (VaR)** and **Conditional VaR (CVaR)**.

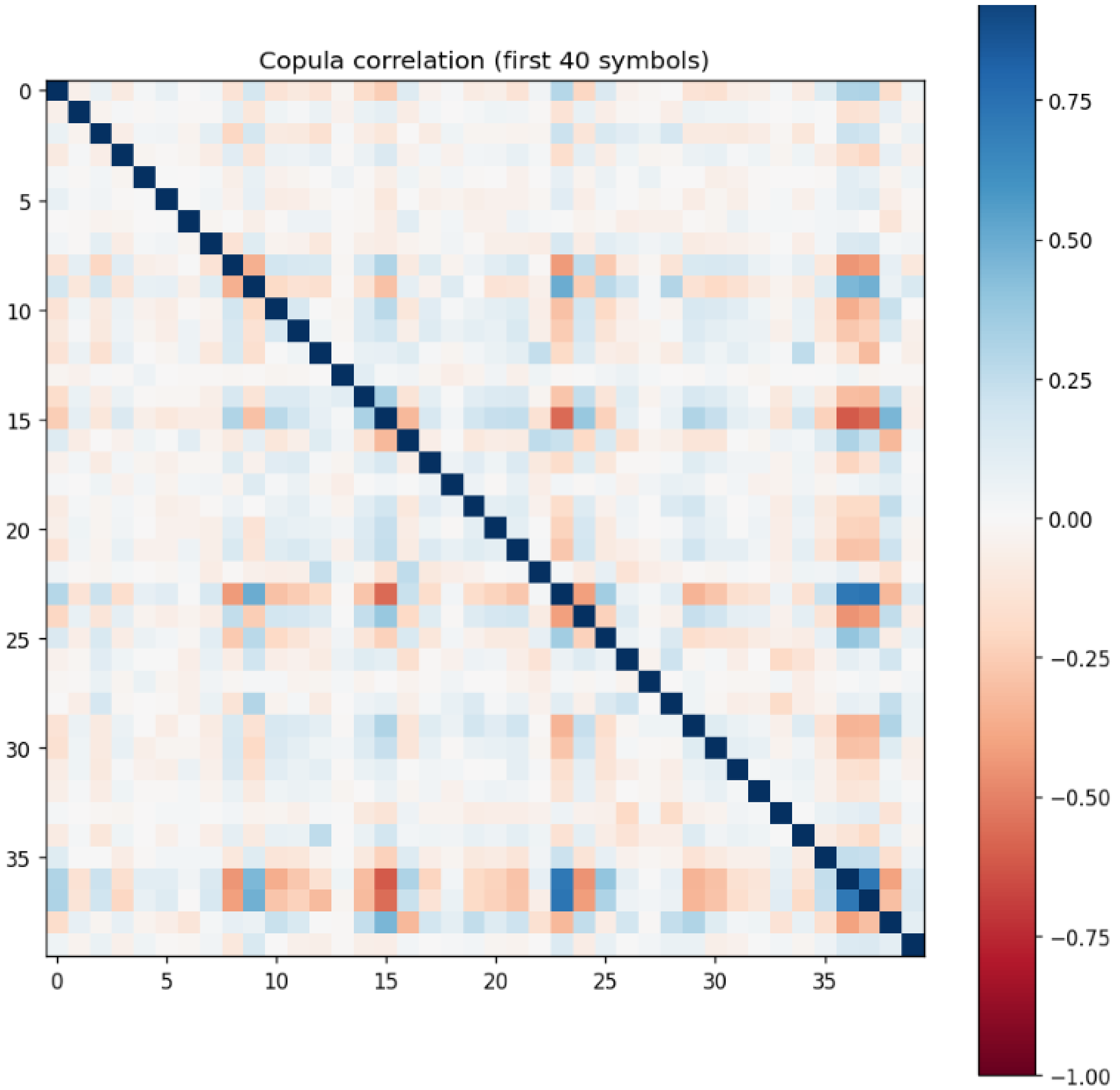
Together, these metrics show both **expected performance** and **potential downside** across hundreds of market scenarios.

Outputs

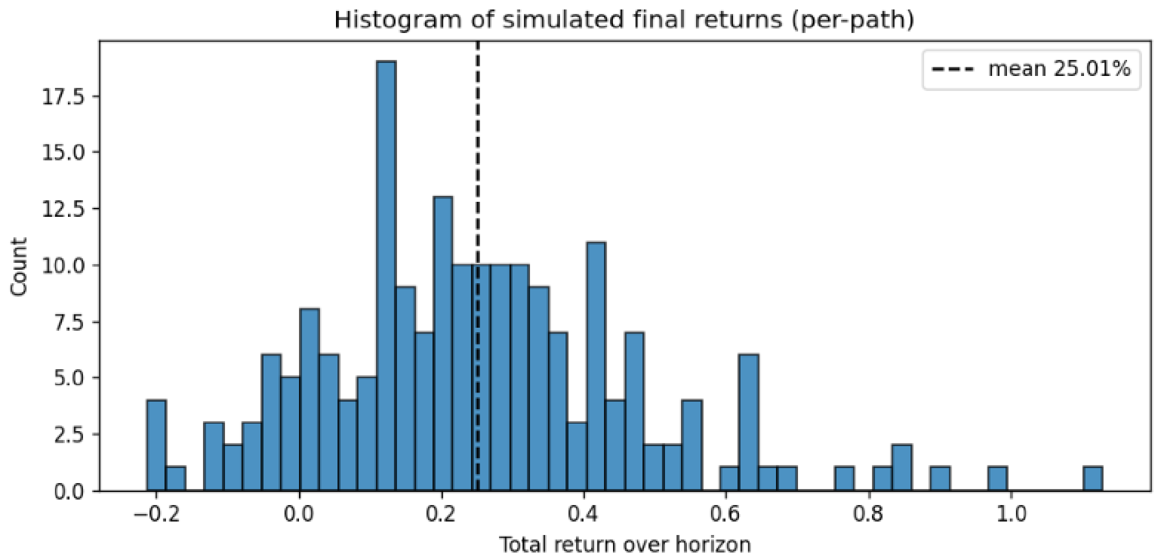
1. The simulator groups stocks into 8 sectors using K-Means based on Correlation patterns.
2. Factor model identified 9 total factors
3. The GARCH(1,1) model captures the volatility
4. Gaussian Copula used to model cross-asset dependence
5. Displays simulator stats over 200 simulations such as mean **returns, std dev, sharpe ratio.**

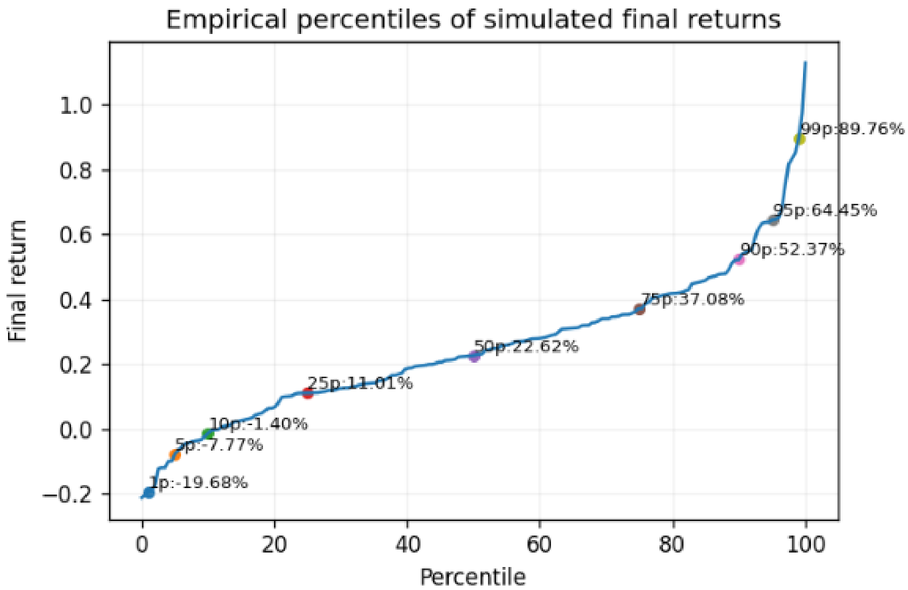
```
Loaded price panel: (2659, 1977) returns: (2658, 1977)
Clusters counts (top 8):
 sector_id
3    442
1    371
2    309
5    248
6    228
7    216
0    142
4     21
Name: count, dtype: int64
Factor names: ['market', 'sector_0', 'sector_1', 'sector_2', 'sector_3', 'sector_4', 'sector_5', 'sector_6', 'sector_7', 'style_mom30']
betas shape: (1977, 10)
residuals shape: (2658, 1977)
GARCH fit: sample factors params keys: ['market', 'sector_0', 'sector_1', 'sector_2', 'sector_3']
Copula fit OK. corr shape: (1977, 1977)
PortfolioSimulator: running engine-driven simulation (GARCH + Copula).
Calling garch_engine.simulate_all_factors(...)
Calling copula_engine.sample(...)

SIM STATS: {'mean': 0.25006675474886814, 'std': 0.22755968475084976, 'sharpe': np.float64(1.0978599297336211), 'sortino': np.float64(6.903889911418226), 'mdd': -0.002111307107835183, 'n_paths': 200, 'horizon': 252}
Sample of simulated final returns (first 10): [ 0.10135671  0.2475148  -0.00568985  0.29906479  0.11343524  0.30758771
 -0.17543619 -0.21285601  0.10707741  0.44739294]
Mean of simulated final returns: 0.25006675474886814
Std of simulated final returns: 0.22755968475084976
Summary appended to: results/simulation_summary.csv
```

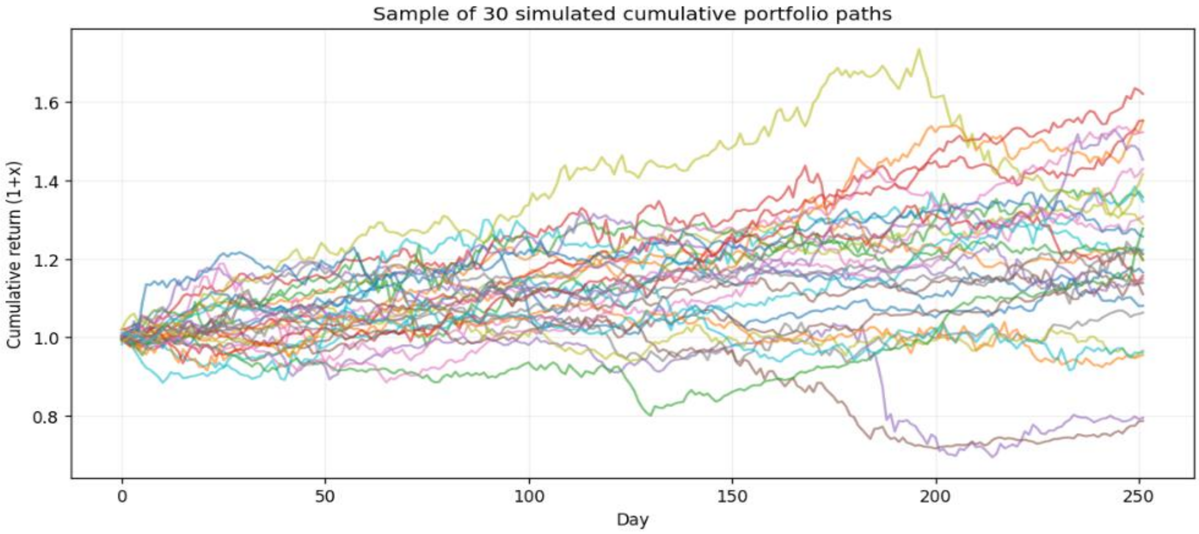
2.Plotted Copula correlation matrix



3.Created histogram of final returns for each path







Takeaways

- I built a full **market simulation engine** that brings together multiple layers of financial behaviour — from factor modelling to volatility, dependence, and risk measurement.
- Learned how to **structure complex systems**: the simulator has five components that interact logically — Factor Model, GARCH Engine, Copula Engine, Regime Model, and Portfolio Metrics.
- Understood that **markets move in patterns, not randomness** — volatility clusters, correlations tighten in stress and returns have memory.

Scope for Improvement

- Introduce **nonlinear dependencies** using t-copulas or vine copulas to capture heavier tails.
- Include **macroeconomic regimes**, like inflationary vs. low-rate periods.
- Add **rebalancing and transaction costs** to simulate active portfolio management.
- Use this simulator to test different **investment strategies** under stress scenarios.