DAT 494 Final Project Report

Ansh Hemang Dani

December 12, 2025

1 Introduction

Portfolio optimization plays a central role in quantitative finance, but its practical implementation is often limited by the reliability of the inputs on which it depends. Classical mean–variance optimization relies on sample estimates of expected returns and the covariance matrix, both of which can be highly unstable when computed from daily asset returns. Even small changes in these estimates can lead to large swings in the resulting portfolio weights, producing solutions that are difficult to interpret and often not robust in out-of-sample settings.

A Bayesian approach provides a natural way to address this instability by treating the unknown return parameters as random variables rather than fixed quantities. By placing priors on expected returns and the covariance structure, we can incorporate regularization directly into the model and obtain posterior distributions that reflect uncertainty in the inputs. This allows us not only to stabilize the estimated parameters but also to generate entire distributions of optimal portfolios instead of relying on a single point estimate.

In this project, I apply a hierarchical Bayesian model to daily returns for eight major U.S. assets from 2019 to 2024. The model uses a multivariate normal likelihood with an LKJ prior on correlations and is fit using Hamiltonian Monte Carlo. After estimating posterior distributions for the return parameters, I evaluate model fit using posterior predictive checks and compare the resulting Bayesian optimal portfolios with those produced by classical mean–variance optimization. The goal is to assess whether the Bayesian framework leads to more stable and interpretable portfolio allocations, especially under realistic levels of return uncertainty.

2 Data

This project uses daily adjusted closing prices for eight actively traded U.S. assets: AAPL, MSFT, AMZN, GOOGL, NVDA, JPM, SPY, and TLT. These assets were selected to provide exposure to large-cap technology stocks, a major financial institution, the broad U.S. equity market, and long-term Treasury bonds. Including TLT adds a useful defensive component because long-term bonds often move inversely to equities, particularly during periods of market volatility.

Historical price data were obtained from the Stooq financial database for the period spanning January 1, 2019 through October 31, 2024. After aligning the trading days across all assets and computing returns, the final dataset contains 1,468 daily observations for each asset.

Daily log returns were calculated using

$$r_t = \log\left(\frac{P_t}{P_{t-1}}\right),$$

which is standard in financial applications because it stabilizes variance and simplifies modeling. No additional preprocessing was required, and the dataset did not contain missing values after alignment. The inherently noisy nature of daily returns makes this dataset a realistic testing ground for evaluating Bayesian approaches to covariance estimation and portfolio construction.

3 Model

The goal of the model is to capture the joint distribution of daily asset returns in a way that accounts for both estimation uncertainty and the strong correlations present among the assets. Classical mean–variance optimization uses sample estimates of the mean vector and covariance matrix, but these estimates can be unstable in finite samples. A Bayesian framework addresses this by treating all unknown parameters as random variables with associated prior distributions.

3.1 Likelihood

Let $r_t \in R^K$ denote the vector of log returns on day t, where $K = 8$ is the number of assets. We assume a multivariate normal model:

$$r_t \sim \mathcal{N}(\mu, \Sigma).$$

Although financial returns may exhibit heavier tails than the Gaussian distribution, this likelihood remains a practical and widely used choice in portfolio theory.

3.2 Priors on Expected Returns

Daily expected returns are difficult to estimate reliably, so we use a hierarchical prior that pools information across assets. We assume

$$\mu_0 \sim \mathcal{N}(0, 1), \qquad \sigma_0 \sim \text{HalfNormal}(0.05),$$

and define

$$\mu_i = \mu_0 + \sigma_0 z_i, \qquad z_i \sim \mathcal{N}(0, 1), \quad i = 1, \ldots, K.$$

This non-centered parameterization improves sampling behavior and shrinks the individual means toward the shared average μ_0.

3.3 Priors on the Covariance Matrix

We decompose the covariance matrix as

$$\Sigma = DRD,$$

where D is a diagonal matrix of volatilities and R is a correlation matrix. The priors are

$$D_{ii} \sim \text{HalfNormal}(0.05), \qquad R \sim \text{LKJ}(\eta = 2).$$

The LKJ prior regularizes correlations away from extreme values while remaining flexible enough to capture realistic dependence among assets.

3.4 Posterior Inference

Posterior inference is performed using Hamiltonian Monte Carlo (NUTS) in PyMC. We ran four chains with 2,000 tuning iterations followed by 1,000 draws per chain, yielding 4,000 posterior samples in total. The non-centered parameterization helped reduce divergences and improved the effective sample sizes across parameters.

3.5 Decision Rule for Portfolio Weights

For each posterior draw $(\mu^{(s)}, \Sigma^{(s)})$, we compute the optimal portfolio of mean-variance

$$w^{(s)} = \Sigma^{(s)-1} \mu^{(s)}.$$

The weights are normalized to sum to one:

$$\tilde{w}^{(s)} = \frac{w^{(s)}}{\sum_{i=1}^{K} w_i^{(s)}}.$$

This produces a posterior distribution of portfolio allocations, which we summarize using posterior means, credible intervals, and distribution plots.

4 Posterior Computation and Diagnostics

Posterior inference for the model was carried out in PyMC using the No-U-Turn Sampler (NUTS), an adaptive form of Hamiltonian Monte Carlo designed to explore posterior distributions efficiently in moderately high-dimensional settings. The parameter space includes the hierarchical mean parameters, the Cholesky factors of the covariance matrix, and the latent variables that govern the correlation structure among the assets.

The final sampling configuration used four chains, each with 2,000 warmup iterations followed by 1,000 draws, producing a total of 4,000 posterior samples. A higher acceptance threshold (`target_accept = 0.95`) and a non-centered parameterization for the mean structure were important in stabilizing the sampler. Earlier attempts produced several hundred divergences, but after reparameterizing and widening the priors on the standard deviations, the number of divergences in the final model run was reduced to just two.

Standard MCMC diagnostics indicate that the sampler performed well. The Gelman–Rubin statistics ($\hat{R}$) were essentially equal to 1 for all parameters monitored, and both effective bulk and tail sample sizes were in the thousands, suggesting adequate mixing across chains. These diagnostics support the reliability of the posterior summaries reported in later sections.

Posterior draws for μ and Σ were extracted from the InferenceData object and used to compute posterior distributions on portfolio weights. Because weight calculation requires inverting Σ, a small ridge term was added when necessary to prevent numerical instability. Summary statistics and visual distributions for the resulting portfolio weights appear in the Results section.

To assess the adequacy of the model, we used posterior predictive checks. Due to limitations in the PyMC version used, the multivariate normal likelihood does not expose elementwise log-likelihood values, so WAIC and LOO could not be computed. Instead, posterior predictive simulations provide the primary means of evaluating how well the model captures the observed return behavior.

5 Results

This section summarizes the posterior distributions for expected returns, volatilities, and correlations, as well as the resulting Bayesian portfolio weights. These results are compared with classical mean–variance optimization, and posterior predictive checks are used to evaluate model adequacy.

5.1 Posterior Expected Returns

Posterior means and 95% credible intervals for the daily expected returns are shown in Figure 1. As expected for daily data, all assets have small means with substantial uncertainty. NVDA and AAPL exhibit the highest posterior means, while TLT shows a slightly negative mean. The hierarchical prior shrinks all means toward a shared center, reducing sensitivity to sampling noise.

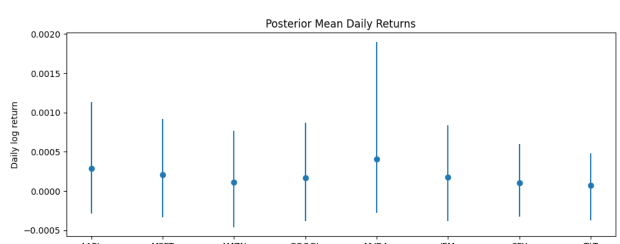

Figure 1: Posterior Mean Daily Returns

5.2 Posterior Correlation Structure

The posterior correlation matrix in Figure 2 reveals several well-known patterns in financial markets. The major technology stocks (AAPL, MSFT, AMZN, GOOGL, NVDA) are strongly positively correlated with each other and with SPY. JPM shows moderate positive correlation with the equity cluster. TLT is negatively correlated with equities, reflecting typical flight-to-quality behavior during periods of market stress.

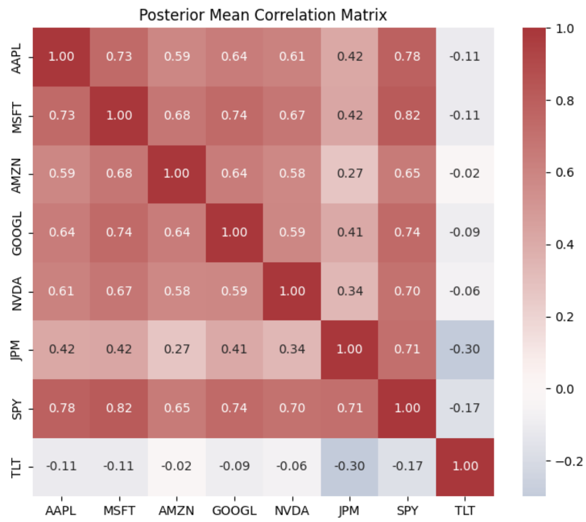

Figure 2: Posterior Mean Correlation Matrix

5.3 Posterior Distribution of Portfolio Weights

Figure 3 shows violin plots of the posterior distribution of optimal portfolio weights. Most distributions are relatively concentrated, but some assets exhibit heavy tails. These extreme values arise from posterior draws in which the covariance matrix is nearly singular, which can inflate the implied weight for certain assets. Despite this, the posterior mean weights are stable and provide an interpretable central summary.

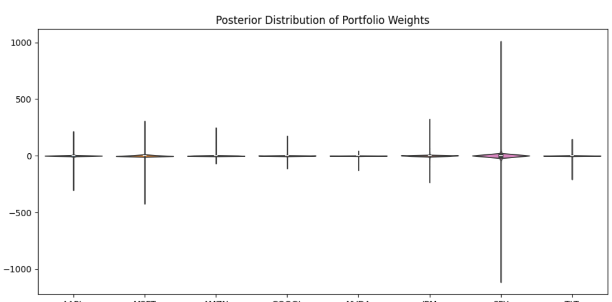

Figure 3: Posterior Distribution of Portfolio Weights

5.4 Comparison with Classical Mean–Variance Optimization

Figure 4 compares Bayesian posterior mean weights with classical mean–variance weights computed from sample estimates. The classical solution allocates extremely large weights to a few assets and negative weights to others, reflecting the instability of sample covariance matrices. In contrast, the Bayesian allocations are more moderate and diversified. Shrinkage in the prior and integration over parameter uncertainty both contribute to reducing extreme allocations.

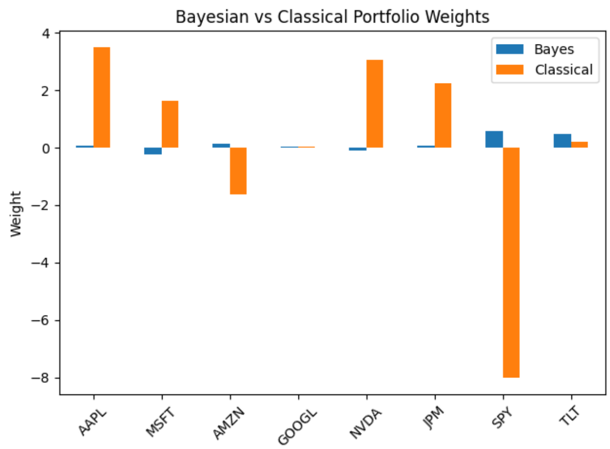

Figure 4: Bayesian posterior mean weights compared with classical mean–variance weights

5.5 Posterior Predictive Checks

Posterior predictive checks provide the primary tool for assessing model adequacy. Density overlays in Figure 5 show that predictive distributions match the observed return distributions well in the central region, although the model underestimates tail thickness for several assets. Time–series predictive intervals in Figure 6 suggest that the model captures most of the variation in AAPL and SPY returns, but it misses a few large shocks, particularly during the COVID period.

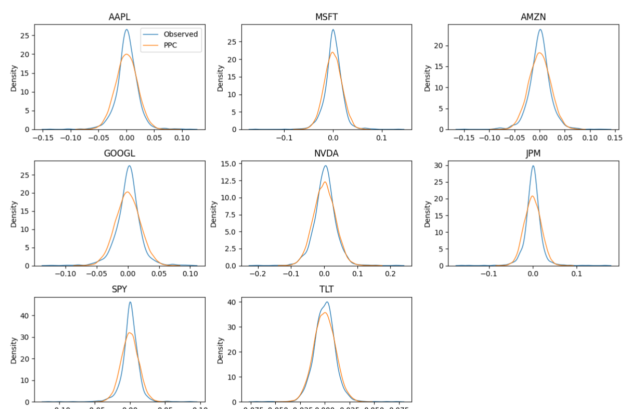

Figure 5: Posterior predictive density overlays for all eight assets

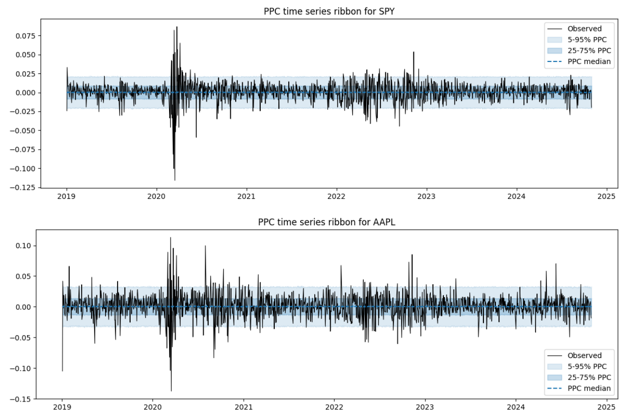

Figure 6: Posterior predictive time–series ribbons for SPY (top) and AAPL (bottom)

Finally, Figure 7 compares observed and posterior predictive means across assets. The model slightly shrinks predictive means toward zero, but overall calibration is reasonable.

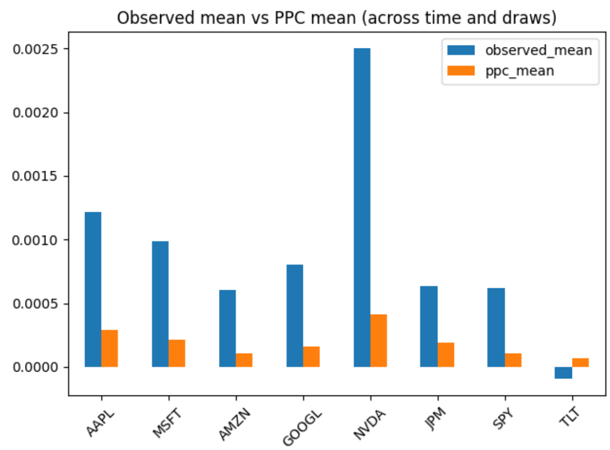

Figure 7: Observed versus posterior predictive mean returns

6 Discussion

The Bayesian framework produced portfolio weights that were more stable and interpretable than those obtained from classical mean–variance optimization. By shrinking noisy return estimates and regularizing the covariance matrix, the hierarchical prior structure helped prevent the extreme allocations that appeared in the classical solution.

The posterior distributions show that the Bayesian approach spreads weight more evenly across assets and naturally incorporates uncertainty through the distribution of optimal weights. This is a clear advantage over classical optimization, which depends heavily on a single sample covariance estimate.

Posterior predictive checks indicate that the model captures the main behavior of the return data, even though the Gaussian likelihood underestimates extreme movements. Overall, the results suggest that Bayesian methods provide a more robust foundation for portfolio construction under realistic levels of uncertainty.

7 Limitations and Future Work

While the Bayesian model performs well overall, several limitations should be acknowledged. The Gaussian likelihood does not fully capture the heavy-tailed nature of financial returns, which causes the model to underestimate large shocks. Using a multivariate Student-t distribution could offer a better fit in such periods.

The model also assumes constant covariance and correlation structures across the entire sample. In practice, market conditions change over time, and a model with time-varying volatility or a factor-based structure could provide additional flexibility.

Finally, pointwise log-likelihood values were not available in the software environment used here, so WAIC and LOO could not be computed. Posterior predictive checks served as the primary validation tool, but future implementations could incorporate model comparison metrics directly.

8 Reproducibility

All data processing, model fitting, and analysis were conducted in Python using PyMC, ArviZ, NumPy, pandas, and other standard scientific computing libraries. The full code for the project is contained in a single script, and all generated outputs (including figures, posterior samples, diagnostic summaries) are saved in the project output directory.

Price data were downloaded directly from the Stooq database at runtime, ensuring that the analysis can be reproduced without requiring additional data files. Running the script from start to finish reproduces the entire workflow, including data collection, posterior inference, posterior predictive checks, and portfolio weight calculations.

The software environment consisted of Python 3.10, PyMC 5, ArviZ, pandas, NumPy, matplotlib, and seaborn. Using similar library versions will allow the results to be replicated closely, with only minor numerical differences possible across systems.

All code used in this project is available at the following repository:

`https://github.com/AnshDani2004/DAT494`