VaR and CVaR Estimation for an Equally Weighted Portfolio of AAPL, NVDA, and MSFT

1. Goal and Rationale

This analysis aims to estimate the daily **Value at Risk (VaR)** and **Conditional Value at Risk (CVaR)** for an equally weighted portfolio comprising Apple (AAPL), NVIDIA (NVDA), and Microsoft (MSFT) over the period from January 1, 2024, to July 1, 2025.

- **VaR** represents the maximum expected daily loss at a given confidence level (e.g., 95% or 99%) under normal market conditions.
- **CVaR** (also known as Expected Shortfall) captures the average loss given that the loss exceeds the VaR threshold, providing a measure of tail risk beyond VaR.

Estimating these metrics helps investors and risk managers better understand potential downside risk and make more informed decisions about portfolio risk management.

2. Data

- **Assets:** Apple (AAPL), NVIDIA (NVDA), Microsoft (MSFT)
- **Period:** January 1, 2024 – July 1, 2025
- **Frequency:** Daily adjusted closing prices
- **Source:** Yahoo Finance, accessed using the Python library yfinance

Real historical price data was used to capture realistic market dynamics, including asset correlations and volatility patterns.

3. Methodology and Tools

We implemented and compared three approaches to estimate portfolio VaR and CVaR:

3.1 Historical Simulation Method

Assumptions: Assumes that the future loss distribution can be approximated by historical return data.
Calculation:

- Portfolio daily returns were computed as the weighted sum of individual asset daily returns.
- VaR at confidence level α% is the (100−α)th percentile of the empirical distribution of portfolio returns.
- CVaR is the average loss of returns worse than the VaR threshold.

Tools: numpy.percentile and filtering for tail returns were used in Python.

3.2 Parametric Method (Variance-Covariance Approach)

Assumptions: Portfolio returns are normally distributed; The portfolio's volatility is constant over the period.
Calculation:

- Portfolio standard deviation calculated using the covariance matrix of returns and weights.
- VaR at confidence level α% calculated as:

  $$VaR_a = -z_a \times \sigma_p$$

  where z_a is the standard normal quantile and σ_p is portfolio volatility.
- CVaR calculated using the formula for expected shortfall under normality:
  $CVaR_a = -\sigma_p \times \frac{\phi(z_a)}{a}$ where ϕ is the standard normal probability density function.

Tools: numpy for matrix operations, scipy.stats.norm for quantiles and PDF.

Note: The Shapiro-Wilk test rejected normality for portfolio returns, indicating this method's assumptions do not hold well, potentially biasing risk estimates.

3.3 GARCH-Based Method

Assumptions:

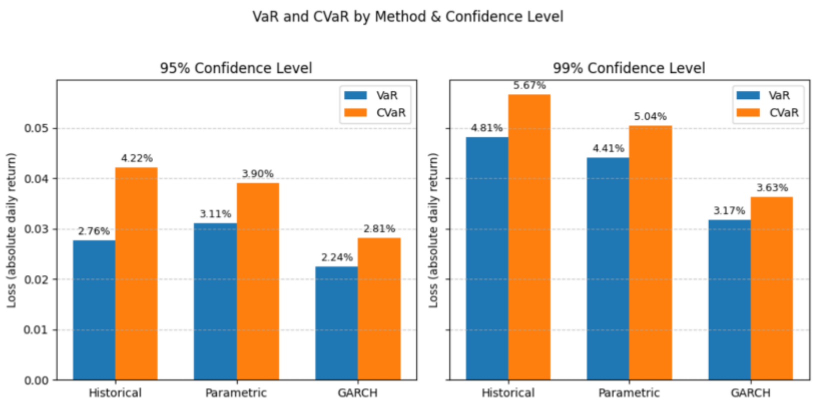
- Returns exhibit volatility clustering; variance changes over time.
- Residuals are conditionally normal (or follow a chosen distribution in advanced models).

Calculation:

- Estimated time-varying volatility using a GARCH(1,1) model fit to portfolio returns.
- VaR and CVaR calculated using the conditional volatility estimates with standard normal quantiles, similar to the parametric method but incorporating changing volatility.

Tools: The arch Python package was used to fit the GARCH model and forecast conditional volatility.

4. Key Results and Takeaways



- The **historical method**, free from distributional assumptions, shows more extreme tail risk (larger magnitude VaR and CVaR) compared to the parametric and GARCH methods, indicating fatter tails in the empirical data.
- The **parametric method**, assuming normality, underestimates tail risk at the 99% level, consistent with the rejection of the normality assumption (confirmed via Shapiro-Wilk test).
- The **GARCH method** captures volatility clustering but tends to give less conservative risk estimates, possibly smoothing extreme movements.

5. Conclusion and Recommendations

- Given the non-normality of returns, the **historical simulation method** provides the most realistic risk assessment for this portfolio.
- The parametric method is simpler but can be misleading if normality assumptions are violated.
- GARCH models add nuance by accounting for volatility dynamics, but may require enhancements (e.g., fat-tailed distributions) to better capture extreme risks.
- For robust risk management, a combination of methods with clear communication of assumptions is recommended.