Building S&P 500 Portfolios with Python

Bojue Wang

July 1, 2025

Background & Motivation

- S&P 500 stocks as building blocks for portfolio construction
- Quantifying risk/return tradeoffs
- Use of Python for Monte Carlo simulation and analysis

Workflow Overview

1. Fetch data (Yahoo Finance)
2. Calculate returns, volatility, Sharpe ratio
3. Monte Carlo simulate ten thousands of random portfolios
4. Visualize results: distribution, risk-return

Data & Preprocessing

- Data: 3-year daily prices for all S&P 500 tickers (allowing up to 10 missing days)
- Calculated annualized return, volatility, and Sharpe ratio for each stock

Core code:

```
import yfinance as yf
tickers = [...]  # S&P500 tickers
data = yf.download(tickers, period='3y', group_by='ticker')
```

Portfolio Simulation

- Random portfolio weights (Dirichlet distribution)
- For each portfolio: Compute annualized return, volatility, and Sharpe ratio
- Repeat for thousands of samples

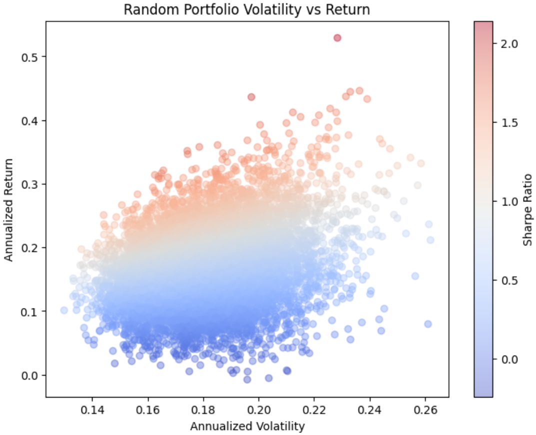
Core code:

```
weights = np.random.dirichlet(np.ones(n_stocks))
annual_return = np.sum(weights * mean_returns) * 252
annual_vol = np.sqrt(weights.T @ cov_matrix @ weights) * np.sq
sharpe = (annual_return - risk_free_rate) / annual_vol
```

Portfolio Return/Risk Distribution

- Distribution of simulated portfolio returns and volatilities
- Identification of extreme high/low-risk portfolios

Risk-Return Tradeoff



- Each point: a random portfolio

Conclusions

- Clear risk-return tradeoff observed
- High-risk portfolios offer greater potential rewards and volatility
- Framework supports further optimization or live-trading extensions

Portfolio Constituents (Summary)

High-Risk:

- Metrics: Return 0.420, Volatility 0.269, Sharpe 1.41
- Major weights: TSLA, AXON, RVTY, EW, NVDA (each $>$ 10%)
- Others: MRK, PLTR, CF, etc. (see appendix)

Low-Risk:

- Metrics: Return 0.115, Volatility 0.130, Sharpe 0.58
- Major weights: KVUE, ABBV, VZ, ELV, MSI (each $>$ 9%)
- Others: ZTS, ICE, NTRS, etc. (see appendix)