Portfolios: Rebalancing

Kevin Crotty
BUSI 448: Investments

Where are we?

Last time:

- Short-sales constraints

Today:

- Simulation
- Investing over multiple periods
- Rebalancing

Simulation

General steps

1. Identify one or more random inputs
2. Set up a function that generates a random draw of the inputs and does some calculations to produce output(s).
3. Run the function in step 2 many times to collect the simulated distribution of the output(s).
4. Summarize the output distribution in some way (average value, percentiles, etc.)

Simulating a single return series

```
from scipy.stats import norm
N_SIMS=1000
sims = pd.DataFrame(dtype=float,columns=['avg_ret'],index=np.arange(N_SIMS))
for s in sims.index:
    rets = norm.rvs(loc=MN, scale=SD, size=T)
    sims.loc[s] = np.mean(rets)
```

Using a function for a single realization

```
# Function to run a single realization
def sim_calc(mean, sd, n_time):
    rets = norm.rvs(loc=mean, scale = sd, size=n_time)
    return np.mean(rets)

# Collect N_SIMS runs of the simulation function
sims = pd.DataFrame(dtype=float,columns=['avg_ret'],index=np.arange(N_SIMS))
for s in sims.index:
    sims.loc[s] = sim_calc(MN,SD,T)
```

Simulating multiple return series

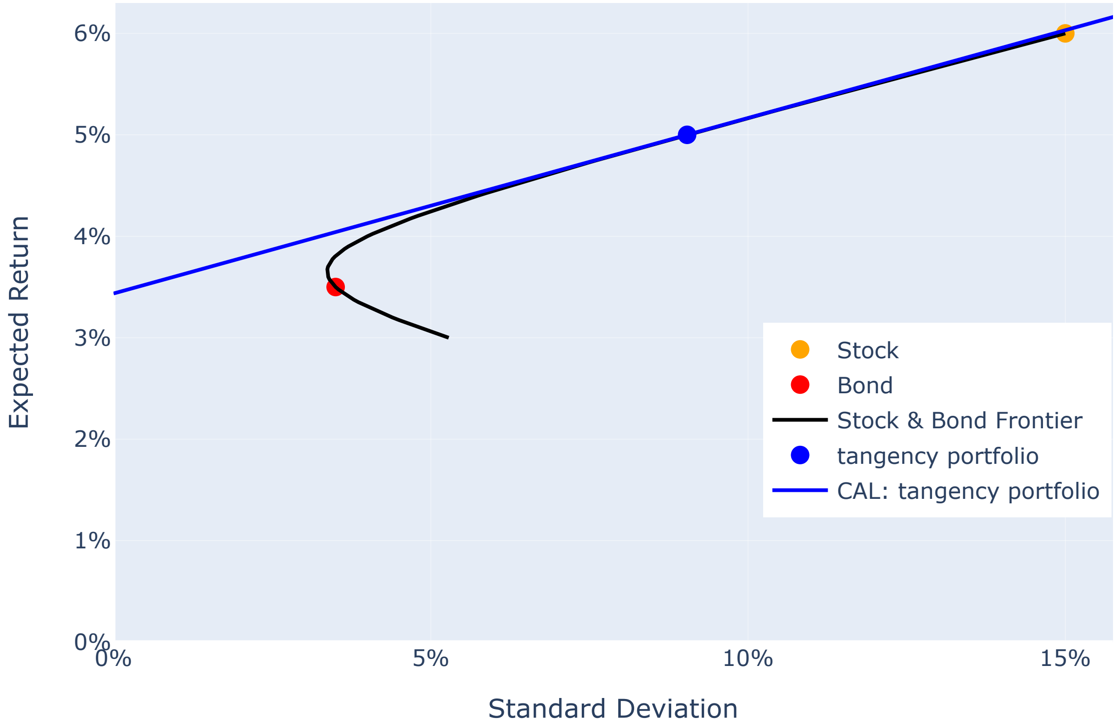
```
# Simulate a single realization and do calculation(s)
def sim_calc(means, cov, n_time):
    n = len(means)
    rets = pd.DataFrame(data=mvn.rvs(means, cov, size=T),
            columns=['ret' + str(i+1) for i in np.arange(n)])
    x = rets.corr()
    corr12 = x.loc['ret1', 'ret2']
    corr13 = x.loc['ret1', 'ret3']
    return corr12, corr13

# Collect N_SIMS runs of the simulation function
sims = pd.DataFrame(dtype=float,columns=['corr12', 'corr13'],
        index=np.arange(N_SIMS))
for s in sims.index:
    sims.loc[s,['corr12','corr13']] = sim_calc(MNS,COV,T)
```

Let's look at a notebook that simulates some of our input statistics for portfolio optimization.

Rebalancing

60-40 Stock-Bond Portfolio



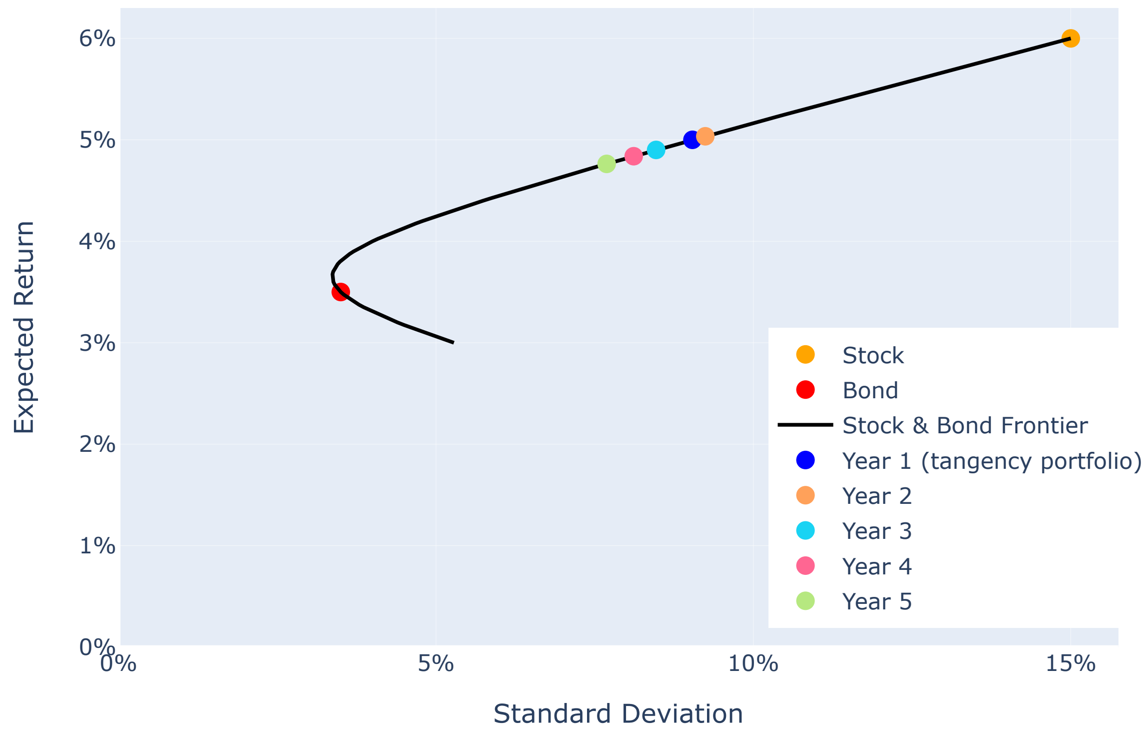
What if we leave this portfolio alone?

- Suppose returns for the stock fund are 12.3% over the first year and the bond fund returns 5.9% over the same period.
- Weights become: 61.4% and 38.6%
- The new weight for asset i is:

$$w_{i,t+1} = w_{i,t} \frac{1 + r_{i,t}}{1 + r_{p,t}}$$

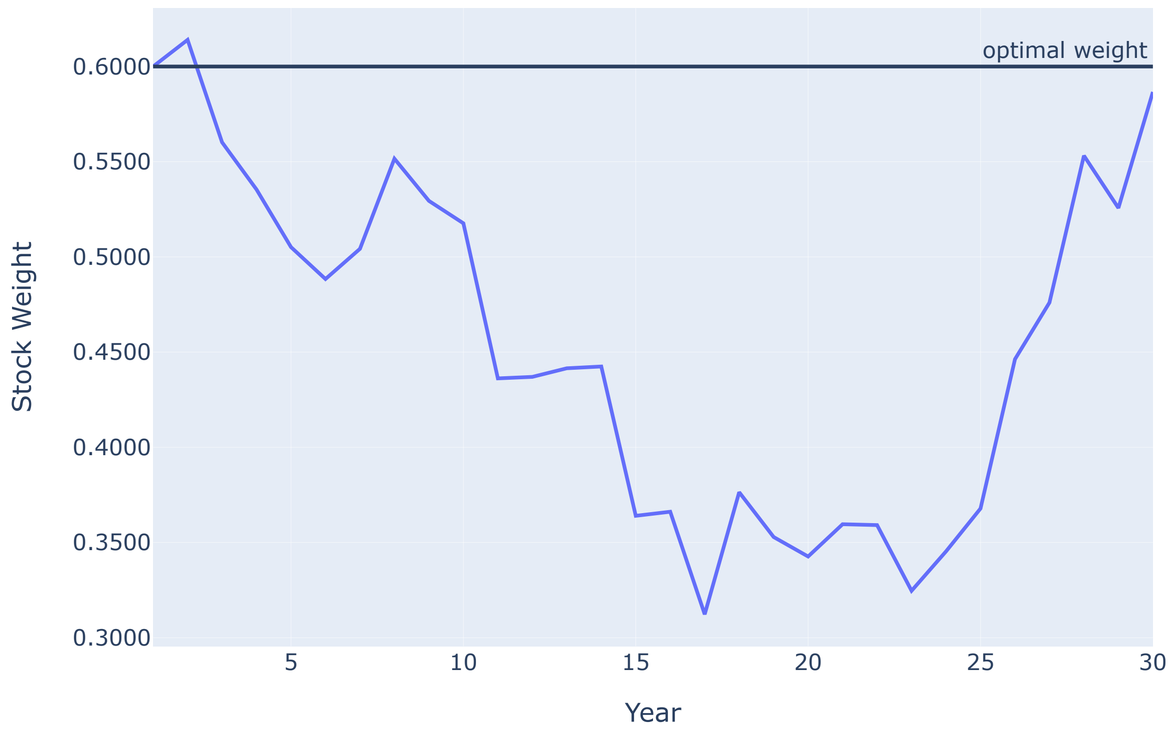
where $r_{p,t} = \sum_{j=1}^{N} w_{j,t} r_{j,t}$ is the time t realized portfolio return for an N asset portfolio

Five years of possible returns



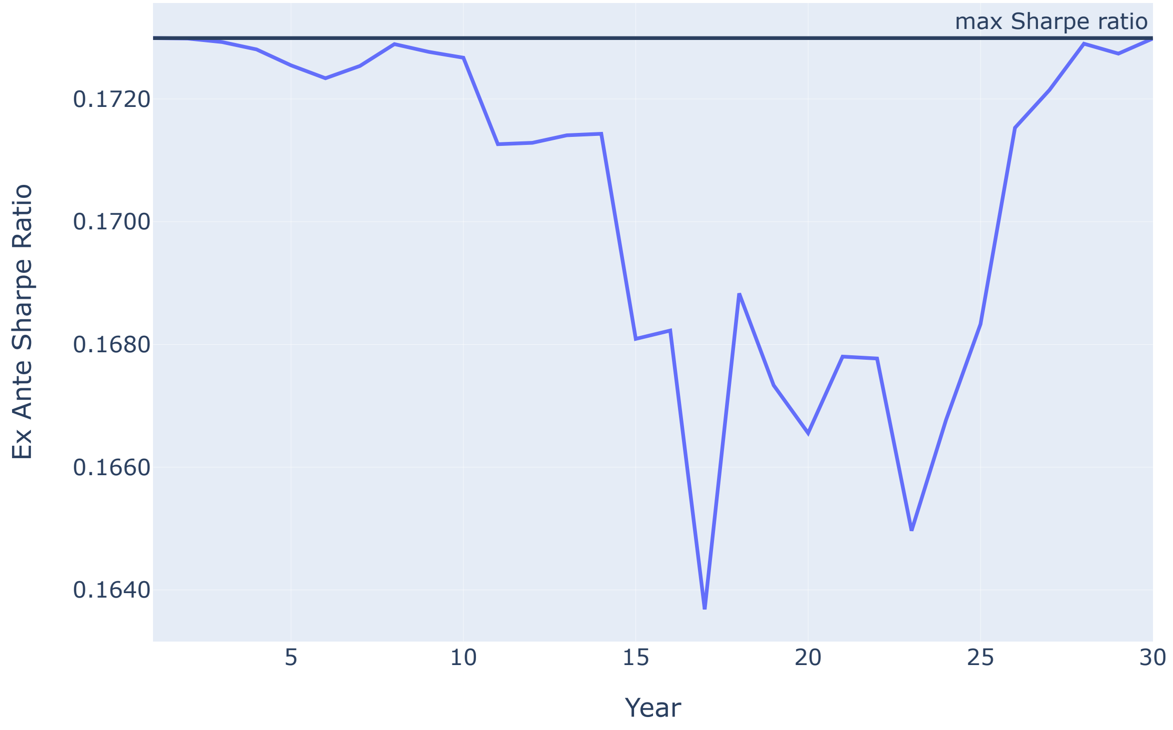
RICE | BUSINESS

30 years of possible returns



RICE | BUSINESS

Ex Ante Sharpe Ratios



Realized Sharpe Ratios

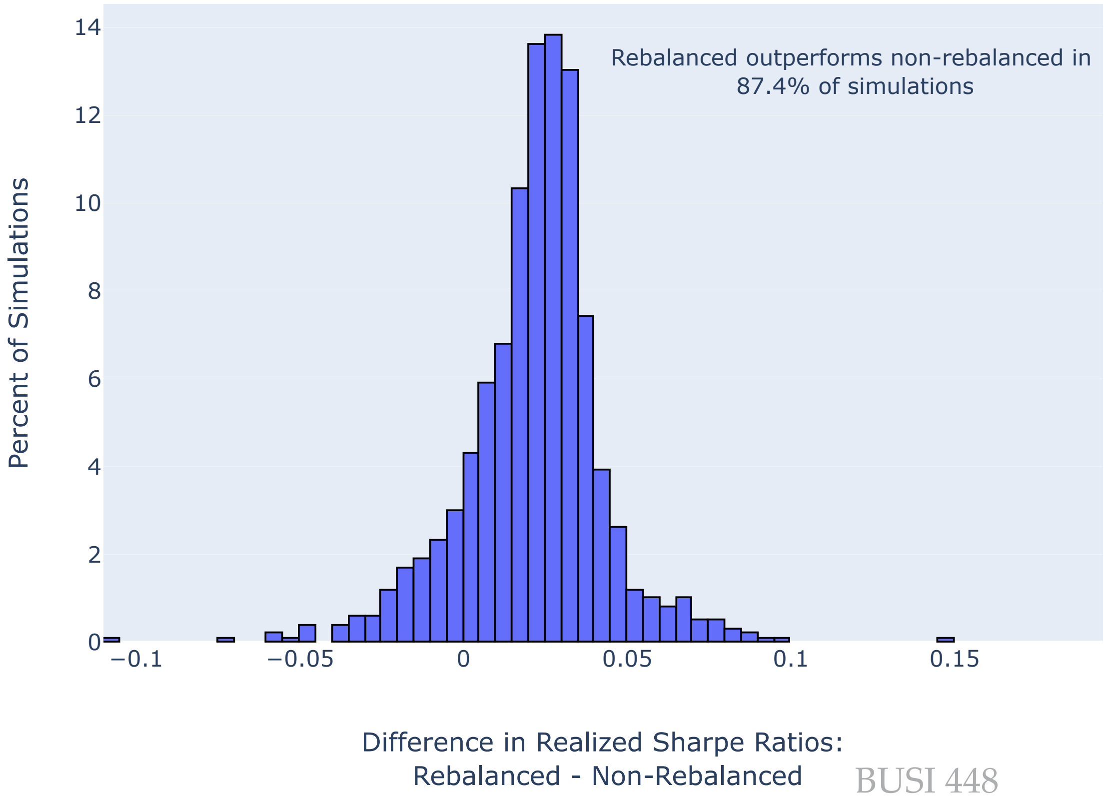
The realized Sharpe ratio of a strategy is its realized average excess return scaled by its realized standard deviation:

$$SR = \frac{\overline{r_p - r_f}}{\mathrm{sd}[r_p - r_f]}$$

For the particular returns above, the realized Sharpe ratio of rebalancing to 60-40 is -0.0277 versus -0.09 for the non-rebalanced portfolio.

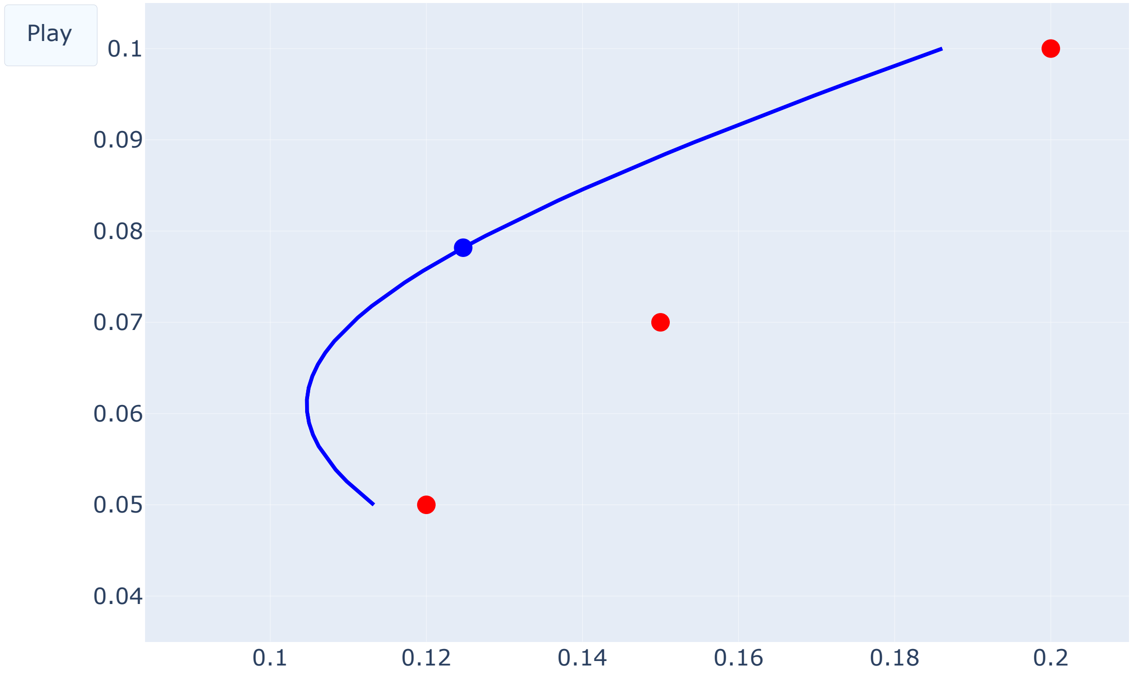
Realized Sharpe Ratios

What if we ran 1000 versions of the 30 year investment period?



RICE | BUSINESS

Rebalancing: 3 assets



Assumptions

We have been assuming:

- returns are independently and identically distributed each period
- the risk-free rate is constant each period.

This means the tangency portfolio is optimal each period.

If expected returns are mean-reverting, then it is also advantageous to rebalance.

Practical Issues: Taxes

- Rebalancing is a contrarian strategy: sell winners and buy losers.
- Selling winners may result in capital gains taxes
- One must weigh the potential benefit of improved portfolio allocation vs. the potential tax exposure of selling overweighted assets

Practical Issues: Transactions Costs

- Similarly, trading may result in fixed or variable transactions costs
- One must weight the potential benefit of improved portfolio allocation vs. the expected costs of the rebalancing transactions

For next time: Input Sensitivity