

Gustav's blog

Random things, mostly about technical stuff

Calculating volatility of multi-asset portfolio, example using Python

A standard way of measuring the risk you are taking when investing in an asset, say for instance a stock, is to look at the assets *volatility*. This can easily be calculated as the standard deviation of the daily returns of the asset. If we for instance have invested all of our money in Apple and we have downloaded the historical price of the stock we could do like this (example needs [Numpy](#) to run):

```
import numpy as np
stock_prices = <Apple's historical stock price>
normalized_prices = np.asarray(stock_prices) / stock_prices[0]
daily_ret = [0.0]
for i in xrange(1, len(normalized_prices)):
    daily_ret.append(normalized_prices[i] / normalized_prices[i-1] - 1)
volatility = np.std(daily_ret)
```

This is all nice and easy when we are only looking at a single asset, in this case Apple. But if you are a bit more serious about your investments you probably understand the importance of *diversifying* your investments and hold a *portfolio* containing several stocks and/or other assets.

By diversifying your portfolio you can lower the volatility of the portfolio and, at least in theory, create a portfolio with lower volatility than any of the individual assets in the portfolio.

So assume for instance that our portfolio consists of three stocks; Microsoft, Apple and Kraft. Assume further that the the weight of the three stocks in our portfolio is 0.3, 0.5 and 0.2. Meaning that 30% of our money is invested in Microsoft, 50% in Apple and 20% in Kraft.

What is now the volatility of the whole portfolio? The naive way would be to take the weighted average of the volatility of the individual stocks. So the volatility of our portfolio, $Vol(p)$, would then be calculated as:

$$Vol(p) = (0.3 * Vol(Microsoft) + 0.5 * Vol(Apple) + 0.2 * Vol(Kraft)) / 3$$

But this is wrong, dangerously wrong. What this method misses to take into account is the *correlation* between the stocks. Correlation tells us how the stocks move in relation to one another, both in terms of direction and of intensity. Correlation between two assets is given as a number between -1 and 1. If the correlation is 1, the two stocks move in perfect sync, if one of them gains 2% the other one will also gain 2%. If one of them falls 5%, the other will also fall 5%.

If the correlation is -1 they move in perfect sync but opposite each other. So when one of the stocks gains 3% the other falls 3%.

A correlation of zero means that there is no relation between how the two stocks move.

So a diversified portfolio should consist of assets that do not correlate "too much". In our three-asset example we can assume that Microsoft and Apple have a strong positive correlation since they are in the same area of business. So adding one of them does not help much with diversification of the portfolio.

Our measurement of volatility should therefore take into account the correlation between each of the assets. The equation for this volatility gets quite hairy for portfolios larger than two or three assets, but fortunately for us we can use a matrix operation for the calculation. If we put the weights of the assets in the portfolio in an array w , and calculate the correlation between each asset in a matrix $corr_matrix$, the variance of the portfolio's daily returns can be expressed as:

$$\text{Var}(p) = w.T * \text{corr_matrix} * w$$

From this we calculate the volatility, i.e. standard deviation as

$$\text{Vol}(p) = \text{Sqrt}(\text{Var}(p))$$

In Python, we could do this calculation as follows, assuming we have calculated the daily return arrays for each asset as before and put them in the variable *daily_returns*.

```
daily_returns = [daily_returns_Microsoft, daily_returns_Apple, daily_returns_Kraft]

# create the correlation matrix
corr_matrix = np.corrcoef(daily_returns)

# portfolio weights
w = np.array([0.3, 0.5, 0.2])

# portfolio volatility
portfolio_volatility = np.sqrt(w.T.dot(corr_matrix).dot(w))
```

Like 0

Post

This entry was posted in Uncategorized on 2 June, 2013

[<https://www.arngarden.com/2013/06/02/calculating-volatility-of-multi-asset-portfolio-example-using-python/>].

2 Comments

 Login ▼

G

Join the discussion...

LOG IN WITH

OR SIGN UP WITH DISQUS 

Share

Best

Newest

Oldest

V

Viral

8 years ago

Xnp0 is correct. Covariance matrix, not correlation

this is wrong, dangerously wrong :p

0 0 Reply **Рэндом Хиро**

8 years ago edited

your volatility formula is wrong, it should be covariance matrix instead of correlation matrix

0 1 Reply [Subscribe](#)[Privacy](#)[Do Not Sell My Data](#)