Assignment 1

Advanced Investments

January 13, 2016

The aim of this assignment is to develop an understanding for the intuition behind mean-variance portfolio allocation. In order to do this we look at a set of country-level equity indexes in our empirical exercises.

Q1: (*Returns statistics*) Download the price levels for the G8 equity indices provided in the 'Assignment 1' folder of the course on Hub.

Compute daily (log) returns and summary statistics for each index. Calculate mean, standard deviation, skewness, kurtosis, autocorrelation of returns, and the correlation matrix. Evaluate whether it is reasonable to approximate the return process by i.i.d. Gaussian distribution.

Q2: (*MV analysis #1*)

In the traditional mean-variance approach the user inputs a complete set of expected returns and the portfolio optimizer generates optimal portfolio weights. Matlab provides the routine '*quadprog*' for numerically solving quadratic optimisation problems with the objective function:

$$\min \frac{1}{2} x' H x + f' x \qquad s.t. \quad Ax \leq b$$

with the option to include additional equality and inequality constraints. Using the course notes, map the Markovitz solution to this routine and solve for the standard portfolio weights w. Plot the annualised historical returns for your test assets along with the mean-variance efficient frontier and mark the tangency portfolio (maximum Sharpe ratio).

Notice, how extreme are the weights. Redo the exercise introducing 'no short-sale constraint' (i.e. restriction on non-negative weights).

Q3: (*MV analysis #2*) Repeat the previous exercise but now use a rolling window of 5 years from end-2004 to today. Thus the first portfolio is based on data from 2000-2004 (inclusive), then 2001-2005, etc. until end-2015. Compute only one portfolio at the end of every year. Is this portfolio significantly time-varying?

Finally, if we compare vs. market cap-based weights how similar are Mean-Variance weights?

1 Appendix: Imposing Short selling constraint in MatLab

To apply no short selling constraint, you can specify the quadratic optimization command 'quadprog' as follows:

```
% Find the minimum variance portfolio
H = 2.0 * Sigma;
f = zeros(N,1);
% 1st Constraint: weights sum to one
Aeq = ones(1,N);
beq = 1.0;
% 2nd Constraint: no short sales
A = -eye(N);
b = zeros(N,1);
[weightsP; sigmaP2] = quadprog(H, f, A, b, Aeq, beq, [], [], [], options)
```

To allow short selling, just change the input 'A' and 'b' in 'quadprog'. For example

```
H = 2.0 * Sigma;
f = zeros(N,1);
% 1st Constraint: weights sum to one
Aeq = ones(1,N);
beq = 1.0;
% 2nd Constraint: no short sales
A = zeros(N);
b = zeros(N,1);
[weightsP; sigmaP2] = quadprog(H, f, A, b, Aeq, beq, [], [], [], options)
```