FINANCIAL ECONOMETRICS AND EMPIRICAL FINANCE - MODULE 2

Exercise Class 1 - February 2012

Asset Allocation in the 70s'

SOLUTION

1. The solution to the first question is quite straightforward; with the lines of code:

```
[filename,pathname]=uigetfile('*.xls');
[data,textdata,raw] = xlsread(filename,1);
```

You can load data from the excel spreadsheet into the Matlab workfile. You will now have three objects in the workfile: the matrix *raw* (which shows time series, headers and dates exactly as they appear in the excel file), the matrix *data* (collecting the time series), and the matrix *texdata* (collecting headers and dates). Dates are imported from Matlab as text.

A useful command you might now use is *datenum.* This function allows to transform dates writtem as text in serial numbers.

```
date=datenum(textdata(3:end,1),'dd/mm/yyyy');
```

2. To answer this question we have to compute asset returns for each risky asset and then subtract the risk free in order to get excess returns. As a first step we can compute the monthly log risk free rate :

```
lrf_m = log(1+(data(:,2)/(100*12)));
```

Bear in mind that yields are always annualized; that's why we are dividing by 12. We are also dividing by 100 for a matter of scaling (in the xls file 1=1%).

We now turn to the German 10-years bond. The time series that we find on the xls file contains the yield of the bond. We therefore have to transform yields into returns. This can be done easily by computing the duration of the bond and multiplying it by variations in yields. As a first step, we calculate log yields for the bond.

```
ly_10_m = (log(1+(lag(data(:,1))/(100*12))));
```

Again, data are rescaled and annualized. We now have to approximate the duration; a way to do it is to use an approximation that can be derived by assuming that coupons are equal to the yield to maturity (just by using of **annuities**).

dur = ((1-(1+(data(:,1)/(100))).^(-10)))./(1-(1+(data(:,1)/(100))).^(-1));

We then calculate returns as:

lret_b_10_m =(lag(dur).*lag(ly_10_m)-(lag(dur)-1).*ly_10_m);

The excess return is just the difference between the monthly returns of the 10-years bond and the monthly return of the risk free rate.

exlret_b_10 = lret_b_10_m-lrf_m;

For what concerns stocks, log total returns are calculated including both log prices and log dividends (here we just show an example for the German market). Of course, dividend yields need to be annualized.

dy_ger_m = data(:,4)/(100*12);
lret_ger_m = log((p_ger./lag(p_ger))+dy_ger_m);

For the US and UK markets, returns have to be adjusted for the exchange rate; we compute exchange rates as:

r_EUvsDOL = log((data(:,9))./lag(data(:,9)));
r_STRvsDOL = log((data(:,10))./lag(data(:,10)));
r_STRvsEU = r_STRvsDOL - r_EUvsDOL;

In our weird notation, we might not be taking thoroughly into account FX conventions. We therefore specify that: EUvsDOL means 1€ = # .### $. For those of you who master FX trading, we can also say that in this example EUR is the base currency.

We the compute returns in EURO terms of the US and UK stock markets:

lret_us_m = lret_us_dol - r_EUvsDOL;
lret_uk_m = lret_uk_dol + r_STRvsEU;

We get excess returns again by just subtracting the log monthly risk free rate.

3. In order to solve this exercise, you just have to compute the cumulative sum (remember we are working with log returns) of excess returns over the chosen sample. As a first step, you have to select the starting and ending date of the sample and count the number of observations in between them. Thus:

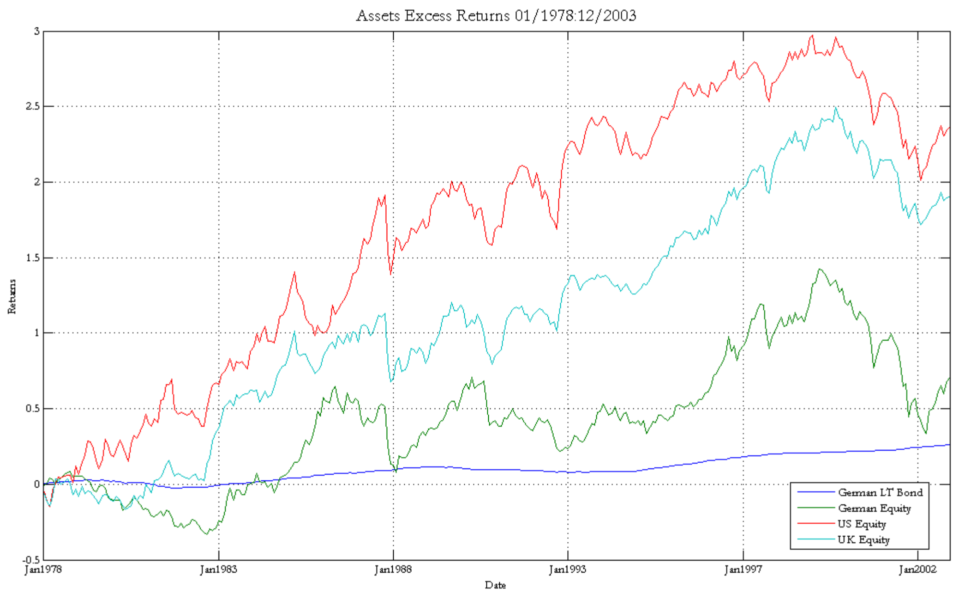
```
s_start = '01/01/1978';
s_end = '01/12/2003';
date_find=datenum([s_start; s_end],'dd/mm/yyyy');
ss=datefind(date_find(1,1),date);
se=datefind(date_find(2,1),date);
```

You can now select the relevant observations in the vectors of excess returns, and collect them into a matrix.

```
R = [exlret_b_10(ss:se) exlret_ger(ss:se) exlret_uk(ss:se)
                     exlret_us(ss:se)];
```

Cumulated performance is derived as: **Perf = cumsum(R);**

This is the result you should get when plotting the time series:



The code for the plot above is the following:

```
figure(1);
plot(Perf);
title(’Assets Excess Returns 01/1978:12/2003’,’fontname’,’Garamond’,’fontsize’,14);
index=1:60:(se-ss+1);
set(gca,’fontname’,’garamond’,’fontsize’,10);
set(gca,’xtick’,index);
set(gca,’xticklabel’,’Jan1978|Jan1983|Jan1988|Jan1993|Jan1997|Jan2002’);
set(gca,’xlim’,[1 (se-ss+1)]);
grid;
ylabel(’Returns’);
xlabel(’Date’);
h=legend(’German LT Bond’, ’German Equity’,’US Equity’,’UK
Equity’,0);
```

A few of brief comments: with the first command you open a new empty figure in Matlab, and set it as figure number 1. In this way, if you open a new figure, you will not overwrite your pervious work. We then plot the matrix **Perf** into the figure. If we are interested in raw results and do not like formatting, our work is done. However, here we discuss some easy commands to improve the appearence of graphs in Matlab. A useful command under this respect is the function **set**, which is used in order to modify figure properties.

In the code, the first thing we do to format the plot is to assign a title and choose its font and size. Then, we assign labels to the x axis. First, we have to state how many ticks we want on the axis and which is the distance of one from the other. That’s why we build the linespace **index** and assign it to the command **’xtick’**. After having chosen the number and position of ticks, we can assign labels, as you can see in row 6 of the code. We then define limits for the x axis, we plot a grid on the chart, we assign axis names and plot the legend.

4. The formal solution for the Markowitz asset allocation problem can be found in the handouts of FINANCIAL ECONOMETRICS AND EMPIRICAL FINANCE - MODULE 1, in section 8. In the answer to question 3, we have to find the weights in the Market Portfolio, for the market composed by the four risky assets. The market portfolio coincides with the **tangency porfolio**, whose vector of weights is formally derived in the notes as:

$$w = \frac{\Sigma^{-1}(\mu - r^f)}{1'\Sigma^{-1}(\mu - r^f)}$$

Where Σ is the historical unconditional variance covariance matrix, μ is the vector of unconditional historical means and r^f is the risk free rate; since we are considering a multiperiod investment horizon, we have to multiply both the unconditional mean vector and the unconditional variance-covariance matrix by

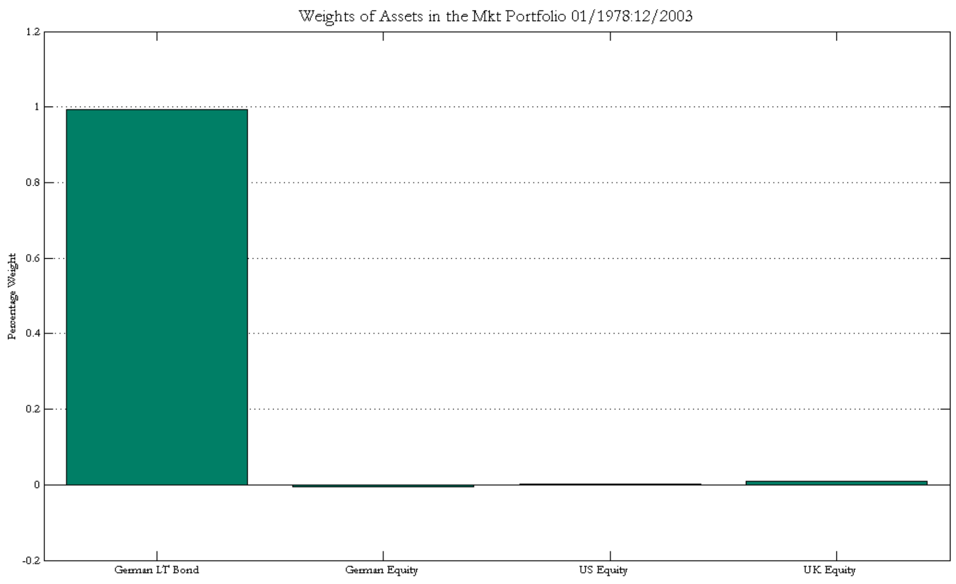
n, where n is the number of periods (months). This is exactly what we do in the code:

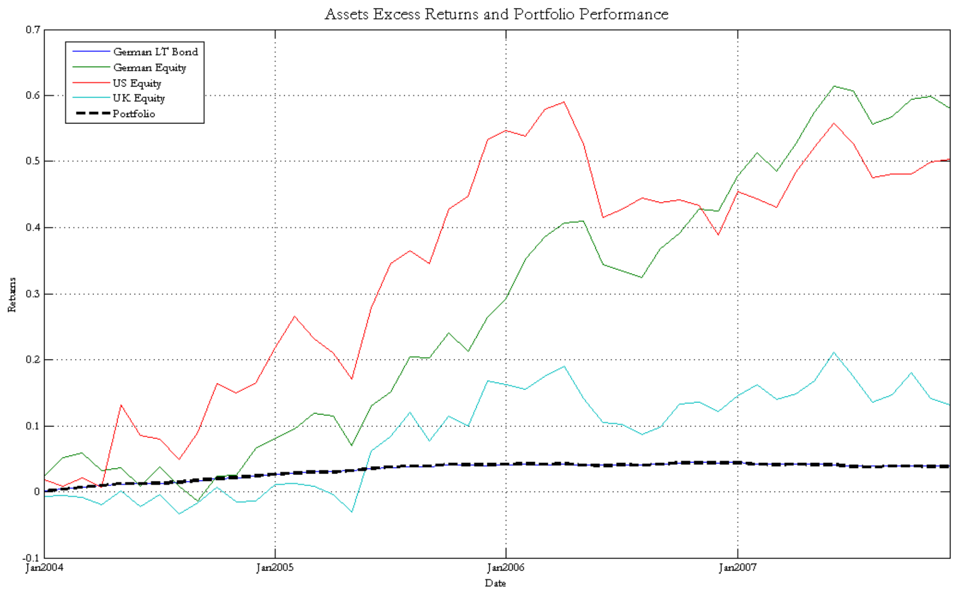
muR = n*mean(R)';
SigmaR = n*cov(R);

Remember that **muR** is a vector of **excess returns**. Then, the vector of weights is:

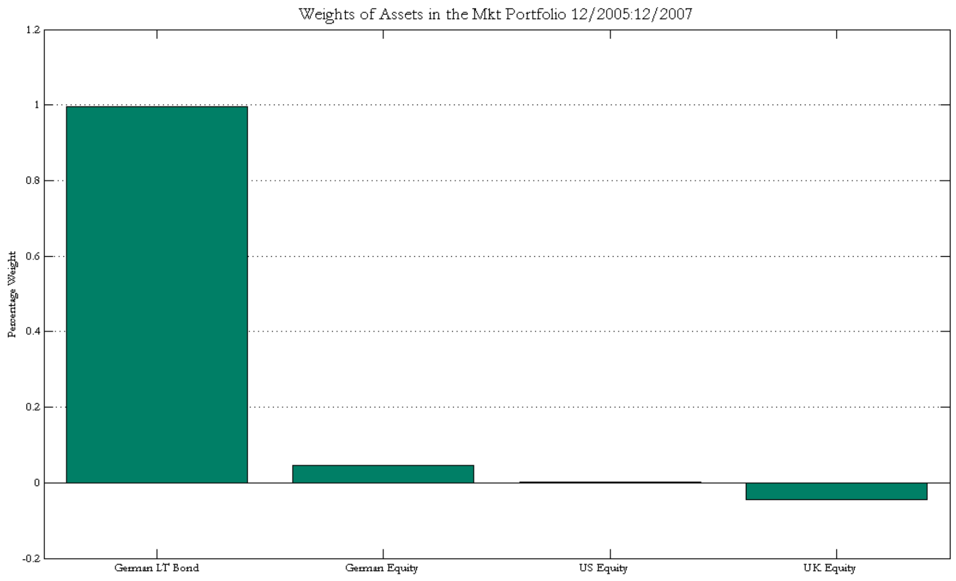
wMP = ((SigmaR^(-1))*muR)./(ones(4,1)'*(SigmaR^(-1)*muR));

5. We now plot the weights and the performance of the portfolio derived using the Markowitz model; as we can see tha portfolio is almost 100% invested in the bond, and over the investment perido dramatically underperforms stocks.





6. We plot weights calculated using data from the sample 01/2004:12/2007; as you can notice their values have slightly changed.



7. You have already met in your studies an asset allocation model able to integrate a view on a certain asset and the Markowitz mean-variance approach. In fact, this solution is discussed in Chapter 12 of the FINANCIAL ECONOMETRICS AND EMPIRICAL FINANCE - MODULE 1 class notes, and happens to be the Black and Litterman model (hereafter B&L). We now propose an (almost) real life implementation exercise of this framework to your asset allocation problem.

As a first step, we have to reverse engineer market expected returns, starting from the historical matrix Σ (for the sample 01/1978:12/2003), the risk aversion coefficient λ and a vector of observed market weights of the assets in our portfolio. We make the assumption that $\lambda = 2.5$, as it is usually done in B&L implementations. For the traded risky assets, we assume that for German investors the market is equally allocated between the 4 investment opportunities; thus each asset will have an hypothetical observed weight of 25% (note that this assumption doesn't affect you final results very much; try and play around with the numbers). We can then compute the implied expected expected excess returns as:

$$\mu^{IMP} - r^f = \frac{1}{\lambda} w^{IMP} \Sigma$$

In the code:

```
wMP_i = [0.25 0.25 0.25 0.25]';
lambda = 2.5;
muMP_i = SigmaR*wMP_i./lambda;
```

As a next step, we have to define the views; here we just say that we expected cumulated excess returns over the whole investment period to be 40% for each stock index. For what concerns the bond, we don't express any view. We therefore specify a selection matrix P:

$$P = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

And a vector of expected values for the views V:

$$V = \begin{bmatrix} 0.4 \\ 0.4 \\ 0.4 \end{bmatrix}$$

To express our confidence in the our insights, we also specify the variance-covariance matrix of the views, and call it Γ. It makes sense to assume that the matrix is diagonal, since it is difficult to make figure out correlation structures between absolute views on different assets (see Chapter 12 for more details). We choose a pretty high value of the standard deviation, which is equal to 0.2 (variance of 0.04) for each one of the equity indexes.

$$\Gamma = \begin{bmatrix} 0.04 & 0 & 0 \\ 0 & 0.04 & 0 \\ 0 & 0 & 0.04 \end{bmatrix}$$

Finally we have to set a parameter which expresses confidence in the estimate based on historical data of the variance covariance matrix of returns. We set this parameter as $\tau = 0.3$, which is again a value in line with what is usually done in applications of B&L.

Then, we write the solution for the weights:

$$s_{BL} = \lambda(\Sigma^{-1}(\mu^{IMP} - r^f) + \Sigma^{-1}K(V - P(\mu^{IMP} - r^f)))$$
$$K = \tau\Sigma P'(\tau P\Sigma P' + \Gamma)^{-1}$$

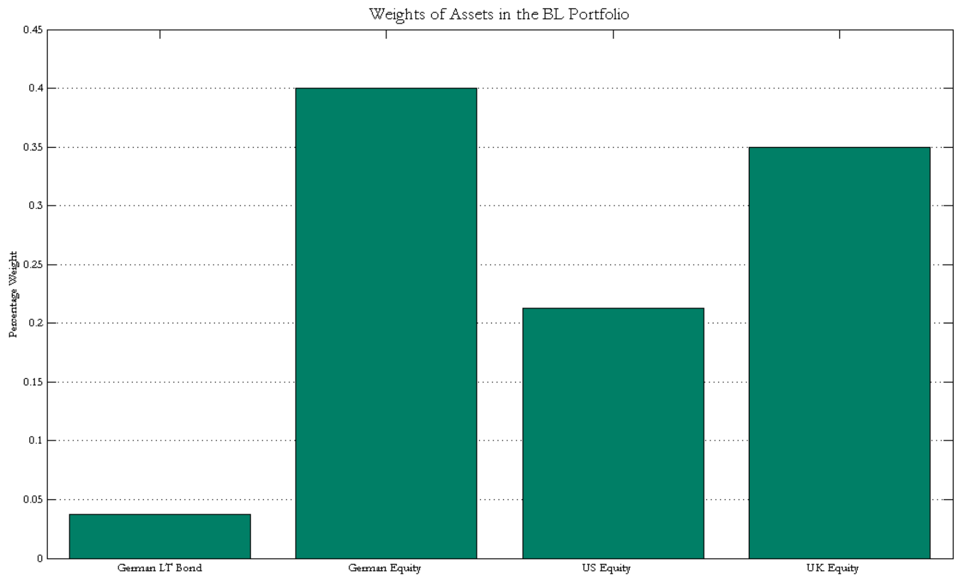
To make the weights sum to one:

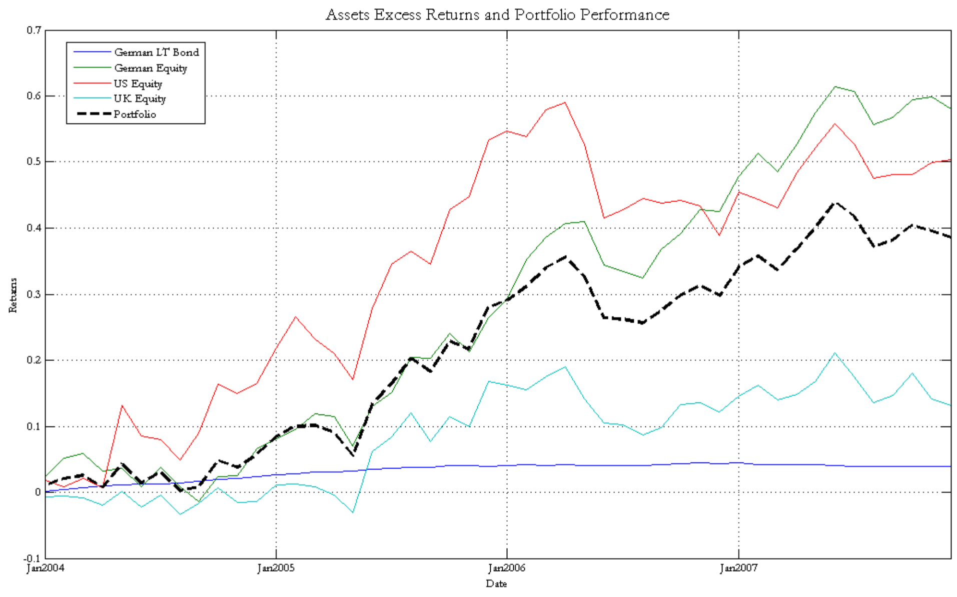
$$w_{BL} = s_{BL}/1's_{BL}$$

In the code:

K = tau*SigmaR*P'*(tau*P*SigmaR*P' + Gamma)^(-1);
sBL = lambda*((SigmaR^(-1))*(muMP_i) + (SigmaR^(-1))*K*(V-P*muMP_i));
wBL = sBL./(ones(4,1)'*sBL);

The plots of weights and performance are reported below:


Weights of Assets in the BL Portfolio
Percentage Weight
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
German LT Bond
German Equity
US Equity
UK Equity


Assets Excess Returns and Portfolio Performance
German LT Bond
German Equity
US Equity
UK Equity
Portfolio
Returns
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-0.1
Jan2004
Jan2005
Jan2006
Jan2007
Date