# Practical 3: Submit solutions in Matlab Grader before Fri 6th Sept at 5pm.

**Bring your lecture notes with you for reference**.

The aim of this practical is to try optimisation on a larger problem, where we need to extract the problem coefficients from a datafile.

Most real-life optimisation problems have many variables and constraints, and it would be far too costly to hand code each of these manually. In this task we will draw data from a file, and then, automatically construct all of the needed matrices and vectors to solve the LP.

**Tasks**

1. Consider the following *portfolio management* problem (from Tutorial 2). A bank has $1 million to invest in variety of bonds offered by the government and other agencies. Each bond has a *rated quality*, and an *after-tax yield*, and a *years to maturity* (how long the investment is committed). The portfolio manager must try to maximise the return on investment, but must also meet other criteria:

   (a) the average quality of bonds cannot be worse than 1.5 (note that for quality, a low number corresponds to high-quality)

   (b) the average years to maturity should not exceed 4 years.

   Assuming there were 4 possible bonds

   (a) What are the variables? *Hint: define variables* $x_1, x_2, x_3$ *and* $x_4$.

   (b) What is the objective?

   (c) Write a series of linear constraints. *Hint: there should be three.*

   (d) What are the bounds on the variables?

   Write your answers out as comments in a Matlab .m file (we will use this in a moment).

2. The actual data for this problem is in the CSV file *portfolio.csv*, that you can obtain from MyUni. The file will be pre-loaded into Matlab Grader.

   The file is a simple CSV (Comma Separated Variable) file, containing a list of possible bonds (one per row of the table), and the relevant data for each. The meaning of each column is described in the first line of the file. **Assume the cost of each bond is $1.**

   Write a Matlab script, using Matlab's `textread` function to read in this data, and construct the appropriate matrices and vectors to solve the problem using `linprog`.

   (a) What proportion of all bonds were included in your portfolio?

   (b) What overall yield do you achieve?

   (c) What average yield do you achieve (as all of the bonds cost the same, the average yield will be the total divided by the number of bonds)?

   Hints

   - There is a *protected* Matlab file `portfolio.p` for you to compare against on MyUni.
   - Use Matlab's `help` utility to find out about `textread`.
   - The template code in Matlab Grader (see following question) might give you some hints on how to solve this problem.

3. **Test your results in Matlab Grader.**

   The skeleton should give you some hints about solving this problem. For instance, you can pass MATLAB's linprog some options created[1] using commands like

   ```
   options = optimoptions(@linprog, Display, off);
   ```

   to reduce the amount of output. See MATLAB Grader for an example with more options set.

   If all goes well, you should almost be able to cut and paste your script from Q2 into the Grader Solution box, and the tests should work.

   Note that Grader is expecting to see the average yield as an output (see the skeleton), not the total yield.

   **Please remember to submit your solution to Grader before the due date.**

[1] This only works for recent versions of MATLAB – older version use a different syntax which you can find using the help function.