TAKE-HOME FINAL EXAM

At `http://sims.princeton.edu/yftp/Times19/TH` you will find an R time series object, loadable from the file `ppi.RData`, consisting of monthly time series of US producer price index data. Its three components are " Processed Goods for Intermediate Demand", "Unprocessed Goods for Intermediate Demand", and "Finished Goods", in that order. Their shortened names in `ppi.RData` are `pts1, pts2, pts3`. (Note that this is somewhat counterintuitive. If you order the indexes by "stage of processing", `pts2` would come before `pts1`.) All three series are indexes (1982=100), monthly, seasonally adjusted. The same data are available at the same directory url as **`.csv`** files, in case you want to use non-R software.

The tasks below are described as using R functions that you have already used for previous exercises, and so should have available. Of course if you use another computational framework you will need to duplicate their functionality.

(1) Estimate a (non-structural) Bayesian VAR for these series, using six lags. If you're using R, the default settings for the prior in `rfvar3` are OK. Use the point estimates to calculate the cointegrating vectors, if any, for the three series.

You can use the procedure described in the cointegration notes in the course second-half web page. This involves taking an LU decomposition of P, the matrix of right eigenvectors of the system matrix formed by arranging the lag coefficients `By[ , , 1:6]` in the first 3 rows of the matrix and **`I`**`, 0` in the last 6 rows. The function `sysmat()`, will form the system matrix, and the function `lu.`**`R`**, available as a text file with the other exam materials, will do the LU decomposition.

About the cointegration notes:

(a) There was an error toward the end of the description of how to find cointegrating vectors that is corrected in a posting today (January 15)

(b) Most of the slides about finding cointegrating vectors deal with what happens when you need to pivot in order to do the LU decomposition. Pivoting is unlikely to be needed with an estimated VAR, and is not needed for this exercise. The simpler procedure described in the paragraph starting with "Suppose that it were possible to form an "LU" decomposition of P" is all that's needed here.

(c) You can work directly with the estimated system matrix, without imposing equality of the non-stationary roots. The algorithm using the LU decomposition finds the $n - q$ linear combinarions of x that explode more slowly than, or shrink faster than, the q biggest roots. For the purposes of this exercise, with T sample size, treat roots differing from one by less than $1/T$ in absolute value as non-stationary, others as stationary. (If there are roots with absolute value greater than $1 - 1/T$ but differing from 1.0+0i by more than 1/T, this whole procedure breaks down.)

Date: January 15, 2020.

(d) The notes discuss the possible need to re-order the columns of P so that the first n columns correspond to the largest n eigenvalues. The columns are already ordered this way in most eigenvalue extraction routines, and certainly in R's `eigen()`

(2) Calculate and plot impulse responses with error bands for the system with triangular orderings. This will of course involve sampling from the posterior. Do it for the "natural" ordering `pts2, pts1, pts3` and also for the reversse ordering. (with `impulsdtrf()` this can be done using the **`order`** argument.)

(3) A Granger causal ordering for a VAR is one in which shocks to higher-ordered variables do not help predict lower-ordered variables. If the "natural" ordering were a complete Granger ordering, the impulse responses of lower-ordered variables to higher-ordered shocks would all be zero, From your plots of error bands, does the "natural" ordering of the variables look like a Granger ordering? Is the reversse ordering clearly not a Granger ordering? Note that it is possible to have a partial Granger ordering, in which the impulse responses are block triangular instead of strictly triangular.