EXERCISE ON IDENTIFICATION THROUGH HETEROSKEDASTICITY.

In this exercise you fit and evaluate a structural VAR, identified through heteroskedasticity, to data on US real GDP, unemployment rate, employment, and employment-to-population ratio. (The denominator of employment/population is civilian, non-institutional population 16 years of age or older.) Data on these four variables is available on the course web site as `yuer.RData` in R data format or `yuer.txt` as text. The text file has the four columns of a 279 by 4 matrix as a single vector, one column after another. The data are quarterly, 1948:I through 2017:III. The GDP and employment data are in logs, the unemployment data are as proportions (not per cent), and the employment ratio data are in percent. The employment ratio data should be divided by 100 before you proceed, so as to get the amount of variation in them the same order of magnitude as for the other series. The `yuer.RData` file contains an R **`ts`** (time series) object, appropriately dated.

You will fit an SVAR, identifying it by assuming the relative variances of structural shocks change just after 1971:I, 1979III, 1982:IV, 2007:IV, and 2012:I.

Software that makes this relatively easy in R is available on the course web site in two R packages, **`optimize`**`.1` and `IDex2019`. These are available as zip files. You can download them and install them as local packages in R. If loaded as packages, they will provide not only the programs, but also help files accessible in R from **`help.start`**`()` or by entering a program name after an initial ? on the R command line.

The program `bvarwrapEx()` calculates the marginal data density conditional on `A`, the matrix of contemporaneous coefficients, and `lmd`, the `n` (number of variables) by `nsig` (number of variance regimes). The inputs it requires are described in the documentation. The main ones for which there are no defaults are

`x`: Contains `A` and the first `nsig - 1` columns of `lmd`, stacked column by column.

`Tsigbrk`: Lists the observation numbers after which new regimes start. The first element is always 0.

`pparams$vprior`: This is a list with elements `sig` and `w`. `sig` is a vector of length the number of variables, giving the scale you expect for residual variances. If the data have been scaled as suggested above, making all elements of `sig` .01 should work.

(1) Find the peak of the posterior density. This can be done by invoking `csminwelNew()` from the **`optimize`**`.1` package, with `bvarwrapEx()` as the function argument. The arguments other than `x` for `bvarwrapEx()` are passed through by giving them as named arguments to `csminwelNew()`. The first 16 elements of `x` are the elements of `A`, which are of the order of 100 and changes in them during the optimization are of the order 1.0, while the last 16 are elements of `lmd`, which change on the order of .01. So the `H0` argument of `csmenwelNew()` should be

```
diag(c(rep(1,16),rep(.01,16)))
```

Date: December 14, 2019.

or something along that line. I found that starting with a scalar matrix for `H0` made convergence slow and erratic. But with the `H0` matrix above I got convergence in a minute or so with a few hundred iterations. My impression, though, is that the posterior density is fairly flat, so parameter vectors that differ in the second decimal place may have nearly identical likelihoods.

(2) Generate draws from the marginal posterior distribution of `(A,lmd)`. This will probably work with random walk Metropolis, and you will of course use `bvarwrapEx()` again to implement the MCMC chain. You'll want to start from somewhere near the peak of the posterior density. Running more than one chain, from different starting points will help you assess convergence. You will save `x` and log posterior density values. (If you use the `verbose=TRUE`, you can save `A` and the full `lmd` matrix instead of the raw `x`.)

Show trace plots and assess whether you have achieved MCMC convergence. You can use the `coda` package, which produces trace plots when **`plot`**`()` is applied to an `mcmc` object and includes the `effectiveSize()` function. The effective sample size should emrge as 100 or more if your results are to be reliable. Usually when effective size is small, you can see from the corresponding trace plot that there is a problem with convergence.

(3) Take a subsample of 1000 `A,lmd` pairs and use them to compute impulse responses to the structural shocks, using the `SVARpostdraw()` function.

An R script that makes the MCMC draws and then generates the irf plots is in `yuexScript.`**`R`**.

Discuss whether the structural impulse responses are well defined (i.e. have narrow enough error bands to be worth interpreting) and whether the shocks have interesting economic interpretations.

(4) Make one or two experiments to check robustness of the results. This could be via adding, subtracting, or moving regime break points given in the default `Tsigbrk` parameter in the script file. Or it could be by varying parameters of the prior. Or it could be testing the fit of the model vs a model (actually a sequence of models) that simply fits a different reduced form VAR, with its own prior, to each regime period (Thus relaxing the constant $A(L)$ assumption.) Or it could be more ambitious — e.g. comparing posterior odds for this vs. a dynamic factor model. Since you would have to create or locate your own tool to fit the dynamic factor model and generate a posterior marginal data density for it, this would definitely be extra credit.