— Problem Set 4 —
Y1: Macroeconometrics
University of Helsinki, February 2012

Antti Ripatti
antti[at]ripatti.net
http://teaching.ripatti.net/metrics/

This problem set contains only a single exercise. It contains several steps. *NOTE! It is useful to rerun your code after each step!*

—Exercise : Estimate the basic New-Keynesian model—

In this problem set you will estimate the parameters of the model of the *Exercise 3 in the Problem Set 2*. I recommend to use Dynare. You have prepared the data in the third problem set. Import that data to Matlab/Octave. Look carefully the instructions in the Dynare User Guide (section 5) and Dynare Manual Section 4.14 (Estimation).

(1) Import the data you prepared in the first exercise of the third problem set. Note that all variables should have the same time span.

(2) Prepare your data by centering all variables, ie compute mean of each variable and subtract the mean from actual data. (There is also similar option in `estimation` command of Dynare, but it seems to contain bugs.)

(3) Consider New-Keynesian model of the *Exercise 3 in the Problem Set 2*.

(4) Add `varobs` section to the code (mod-file) defining what are the observable variables.

(5) You may filter (smooth) the shock processes given the data and the parameters values you have chosen (see page 45 of the Manual). This can be done with the following set of commands

```
estimation(datafile=nkdata,mode_compute=0,mh_replic=0,smoother,order=1)
y pie i d m s ;
figure; plot(oo_.SmoothedVariables.m);
figure; plot(oo_.SmoothedVariables.d);
figure; plot(oo_.SmoothedVariables.s);
```

(6) Try to estimate the parameters of the model by Maximum Likelihood. Note, that in Dynare you tell the initial values of the parameters to the optimizer (here `csminwel`) and possible bounds of the parameter values by the following command (this is an example)

```
estimated_params;
beta, 0.99, 0.00001, 0.999999;
std eps_d, 0.1;
end;
```

This also tells what are the parameters to be estimated! Useful options of the `estimation` command are the following

```
estimation(datafile=nkdata, order=1, prefilter=1, mode_check)
y pie i d m s;
```

You probably will end up to the situation when (minus log) Hessian is not positive definite. This typically results from lack of identification (of parameter(s)). Try to estimate a subset of parameters. What are the ones you are able to estimate.

(7) Turn to Bayesian estimation. Add `estimated_params` section to the code. This section defines prior distribution of the parameters.

Note the following ranges of parameters $0 < \beta < 1$, $\lambda > 0$, $\phi_\pi > 1$, $\phi_y > 0$, $0 < \rho_d, \rho_m, \rho_s < 1$, $\sigma > 0$ in choosing the prior distribution. Standard errors are, of course, positive.

The same `estimation` command options as in the previous set applies here too. Look carefully the plots of the prior distribution. Are they sensible (regarding, for example, the above ranges)!?

Once you are happy with the priors, check the identification by the Dynare command

`identification;` The details of the command may be found from the document: `https://www.ifk-cfs.de/fileadmin/downloads/events/conferences/monfispol2011/RATTO_IdentifFinal.pdf`.

(8) Next starts the *debugging sequence*[1]! This means that lot of problems arrive. You may try the following options of the estimate command:

```
estimation(datafile=nkdata, prior_plot=1, mh_jscale = 0.2, mode_compute=4,
mode_check, mh_replic=1000,smoother,order=1) y pie i d m s ;
```

It is likely that you end up very bad local extrema, where the (minus) Hessian is not positive definite. This is bad situation. Try to use other initial values (for example, those that you calibrated in the problem

[1] Programming always involves debugging. Modeling involves too.

set three). You may also try another optimization algorithm (see the Dynare Manual, option `mode_compute`).

Keep in mind what we talked about acceptation ratio in the lecture. The option that scales the covariance matrix of the proposal distribution is `mh_jscale`. Note that Metropolis-Hastings chains need to be much longer than 1000. For debugging (and for exercise) purposes shorter chains like 10 000 - 100 000 is fine. Compute also the Markov chain diagnostics and interprete them.

(9) (Voluntary!) Once you have estimated the model's parameters distribution. Compare the model moments (computed at median of the parameter values) and data moments again. What do you think of the fit?