Quantitative Macroeconomics

Winter 2023/24

Week 12

Willi Mutschler
willi@mutschler.eu

Version: 1.0
Latest version available on: GitHub

Contents

1. Bayesian Estimation of VAR(p)

Consider the following K-variable VAR(p) model:

$$y_t = c + A_1 y_{t-1} + \cdots + A_p y_{t-p} + u_t = AZ_{t-1} + u_t$$

where $E(u_t) = 0$, $E(u_t u_t') = \Sigma_u$ and $E(u_t u_s') = 0$ for $t \neq s$. Define $\alpha = vec(A)$ and assume that $u_t | y_{t-1}, \ldots, y_1 \sim N(0, \Sigma)$.

1. Explain why Bayesian methods are especially attractive when estimating a VAR(p) model.
2. Assume that the prior for α is normal with mean α_0 and covariance matrix V_0. Provide an expression for the posterior conditional on Σ_u.
3. Assume that the prior for the VAR covariance matrix Σ_u is Inverse Wishart with degrees of freedom v_0 and scale matrix S_0. Provide an expression for the posterior conditional on α.
4. Briefly outline the basic steps of the Gibbs sampling algorithm given the conditional posteriors.
5. Provide intuition behind the "Minnesota prior" and have a look at a possible implementation of it given the file `BVARMinnesotaPrior.m` in the appendix.

Hints

- Use `mvnrnd(alpha1,V1)` to draw from a multivariate normal distribution with mean α_1 and covariance matrix V_1. Make sure your covariance matrix is symmetric: $V_1 = \frac{1}{2}(V_1 + V_1')$.
- Use `inv(wishrnd(inv(S1),v1))` to draw from an Inverse Wishart distribution with degrees of freedom v_1 and scale matrix S_1.

Readings

- Kilian and Lütkepohl (2017, Ch. 5)
- Koop and Korobilis (2010, Ch. 1-2)

2. Bayesian Estimation of a VAR model for the US economy including the Zero-Lower-Bound

Consider a VAR(p) model for the US economy which includes (in this ordering) the federal funds rate, government bond yield, unemployment and inflation. The sample period consists of 2007m1 to 2010m12. Data is given in the file `USZLB.csv`.

1. Estimate the parameters of a VAR(2) model with a constant by using Bayesian methods, i.e. a Gibbs sampling method where you assume a Minnesota prior for the VAR coefficients and an Inverse Wishart prior for the covariance matrix.
   To this end:
   - Define a Minnesota prior for the VAR coefficients. The prior mean α_0 should reflect the view that the VAR follows a random walk. Set the hyper-parameters for the prior covariance matrix V_0 such that the tightness parameters on lags of own and other variables are both equal to 0.5, and the tightness parameter on the constant term is equal to 1.
     Hint: You may want to use the `BVARMinnesotaPrior.m` function in the appendix.
   - Define an Inverse Wishart prior for the covariance matrix with degrees of freedoms v_0 equal to the number of variables and the identity matrix as prior scale matrix S_0.
   - Initialize the first draw of the covariance matrix with OLS values.
   - Draw 40000 times from the conditional posteriors

     $$\begin{aligned} p(vec(A)|\Sigma, Y) &\sim N(a_1, V_1) \\ p(\Sigma|vec(A), Y) &\sim IW(v_1, S_1) \end{aligned}$$

     where

     $$\begin{aligned} V_1 &= \left(V_0^{-1} + ZZ' \otimes \Sigma^{-1}\right)^{-1} \\ a_1 &= V_1 \left(V_0^{-1} a_0 + (Z \otimes \Sigma^{-1}) vec(Y)\right) \\ v_1 &= T + v_0 \\ S_1 &= S_0 + (Y - AZ)(Y - AZ)' \end{aligned}$$

     and keep the last 10000 draws for inference.
     Optionally: check the stability of the draws of the coefficient matrix A, i.e. compute the eigenvalues of the companion matrix and discard the draw if the modulus of all eigenvalues of the companion form is larger than one.

2. As the sample period includes the financial crisis, redo the exercise but now use a small prior variance to reflect the view that monetary policy is at the effective lower bound and hence the federal funds rate is unlikely to respond to changes in the other variables.

Readings

- Kilian and Lütkepohl (2017, Ch. 5)
- Koop and Korobilis (2010, Ch. 1-2)

References

Kilian, Lutz and Helmut Lütkepohl (2017). *Structural Vector Autoregressive Analysis*. Themes in Modern Econometrics. Cambridge: Cambridge University Press. ISBN: 978-1-107-19657-5. URL: https://doi.org/10.1017/9781108164818.

Koop, Gary and Dimitris Korobilis, eds. (2010). *Bayesian Multivariate Time Series Methods for Emprirical Macroeconomics*. Foundations and Trends in Econometrics 3.2009,4. Boston: now. ISBN: 978-1-60198-362-6.

A. BVARMinnesotaPrior.m

progs/matlab/BVARMinnesotaPrior.m

```matlab
function [alpha_prior, V_prior, inv_V_prior, v_prior, S_prior, inv_S_prior] = BVARMinnesotaPrior(Y,const,p,hyperparams)
% [alpha_prior, V_prior, inv_V_prior, v_prior, S_prior, inv_S_prior] = BVARMinnesotaPrior(Y,const,p,hyperparams)
% -----------------------------------------------------------------------
% Outputs Minnesota Prior adapting codes by Gary Koop and Dimitris Korobilis for
% "Bayesian Multivariate Time Series Methods for Empirical Macroeconomics"
% -----------------------------------------------------------------------
% INPUTS
%       - Y           : matrix of data. [number of periods (T) x number of variables (K)]
%       - const       : 0 no constant; 1 constant; 2 constant and linear trend. [scalar]
%   - p           : number of lags. [scalar]
%   - hyperparams : tightness parameters for Minnesota prior on
%                       - 1st value: lags of own variable
%                       - 2nd value: lags of cross variables
%                       - 3rd value: exogenous variables, i.e. constant term, trends, etc
%                   [3x1] vector (optional)
% -----------------------------------------------------------------------
% OUTPUTS
%   - alpha_prior : prior mean for VAR coefficients, reflects view that VAR follows a random walk. [K*(const+K+K*(p-1) x 1]
%   - V_prior     : variance for prior on VAR coefficients. [K*(const+K+K*(p-1) x K*(const+K+K*(p-1)]
%   - inv_V_prior : inverse of V_prior. [K*(const+K+K*(p-1) x K*(const+K+K*(p-1)]
%   - v_prior     : prior degrees of freedom for Inverse Wishart distribution for covariance matrix. [scalar]
%   - S_prior     : prior scale matrix for Inverse Wishart distribution for covariance matrix. [KxK]
%   - inv_S_prior : inverse of S_prior. [KxK]
% -----------------------------------------------------------------------
% Willi Mutschler, January 23 2023
% willi@mutschler.eu
% -----------------------------------------------------------------------

[T,K] = size(Y); % T is number of periods, K number of variables

% initialize prior for VAR coefficients
A_prior     = [zeros(K,const) eye(K) zeros(K,K*(p-1))];
alpha_prior = A_prior(:);

% hyper-parameters on the variance of alpha_prior
if nargin < 4                  % set standard values
    lambda1 = 0.6;
    lambda2 = 0.5;
    lambda3 = 10^2;
else                           % set user-provided values
```

```
    lambda1 = hyperparams(1);
    lambda2 = hyperparams(2);
    lambda3 = hyperparams(3);
end

% get residual variances of univariate p-lag autoregressions with deterministic terms
% these will be used to adjuzst for differences in the units the variables are measured in
sigma_sq = zeros(K,1); % initialize vector to store residual variances
for i = 1:K
    Ylag_i = lagmatrix(Y(:,i),1:p);  % create lags of dependent variable in i-th equation
    if const == 0                    % no deterministic terms
        Z_i = transpose(Ylag_i(p+1:T,:));
    elseif const == 1                % add constant
        Z_i = transpose([ones(T-p,1) Ylag_i(p+1:T,:)]);
    elseif const == 2                % add constant and linear trend
        Z_i = transpose([ones(T-p,1) (p+1:T)' Ylag_i(p+1:T,:)]);
    end
    Y_i = transpose(Y(p+1:T,i));     % dependent variable in i-th equation
    alpha_i = (Y_i*Z_i')/(Z_i*Z_i'); % OLS estimate of i-th equation
    u_i = Y_i - alpha_i*Z_i;         % OLS residual of i-th equation
    sigma_sq(i,1) = (1./(size(u_i,2)-p-const))*(u_i*u_i'); % OLS error variance
end

% create V_pr, an array of dimensions K x (const+K*p),  which will contain
% the diagonal elements of the covariance matrix, in each of the K equations.
V_pr = lambda3 * repmat(sigma_sq,1,const); % prior variances for deterministic terms (if any)
V_i = zeros(K,K); % initialize diagonal elements of prior variance for VAR coefficients
for l = 1:p % for each lag
    for i = 1:K % for each equation
        for j = 1:K   % for each RHS variable
            if i == j
                    V_i(i,j) = lambda1/(l^2); % tightness/variance on own variable
            else
                    V_i(i,j) = (lambda2*sigma_sq(i,1)) ./ ((l^2)*sigma_sq(j,1)); % tightness/variance on cross variables, adjusted for differences in units
            end
        end
    end
    V_pr = [V_pr V_i]; % concatenate with variance on deterministic terms
end
V_prior = diag(V_pr(:));        % now V is a diagonal matrix with diagonal elements V_i
inv_V_prior = diag(1./V_pr(:)); % inverse of a diagonal matrix is just the reciprocate of the values on the diagonal

% hyper-parameters on SIGMA ~ invWishart(v_prior,S_prior)
v_prior = K;                % degrees of freedom equal to number of variables
```

```
S_prior = eye(K);            % prior scale matrix
inv_S_prior = inv(S_prior); % inverse of prior scale matrix

end
```

B. Solutions

1 Solution to Bayesian Estimation of VAR(p)

1. Given the large number of parameters in VARs, estimates of objects of interest (e.g. impulse responses or forecasts) can become imprecise in large models. The Bayesian paradigm enables one to incorporate prior information and generally this makes the estimates become more precise. Moreover, Bayesian methods provide an easy way to characterize estimation uncertainty by looking at the posterior distribution.

2. Conditional on Σ_u and assuming a normal prior for α, the conditional posterior is given by (see the readings for the algebra)

$$p(\alpha|\Sigma_u, Y) \sim N(\alpha_1, V_1)$$

   where

$$V_1 = (V_0^{-1} + ZZ' \otimes \Sigma^{-1})^{-1}$$
$$\alpha_1 = V_1(V_0^{-1}\alpha_0 + (Z \otimes \Sigma^{-1})vec(Y))$$

3. Conditional on α and assuming an Inverse Wishart prior distribution for Σ_u, the conditional posterior is given by (see the readings for the algebra):

$$p(\Sigma_u|\alpha, Y) \sim IW(v_1, S_1)$$

   where

$$v_1 = T + v_0$$
$$S_1 = S_0 + (Y - AZ)(Y - AZ)'$$

4. Gibbs sampling consists of the following steps:
   a) Set priors for the VAR coefficients and the covariance matrix. Set a starting value for Σ_u, e.g. to OLS values.
   b) Compute the moments of the conditional posterior distribution for the VAR coefficients, α_1 and V_1, and take a draw $\alpha(j)$ from $N(\alpha_1, V_1)$.
   c) Draw $\Sigma_u(j)$ from its conditional posterior distribution $IW(v_1, S_1)$.
   d) Repeat steps (b) and (c) a large number M of times to generate sequences $\{\alpha(1), \ldots, \alpha(M)\}$ and $\{\Sigma_u(1), \ldots, \Sigma_u(M)\}$ and use the last L draws for inference.

5. A variety of priors can be used with VAR models, but 3 issues arise:
   a) VAR models are not parsimonious: they have many coefficients to estimate. Ideally our prior should provide information to improve the precisions of estimates by focusing on the most important coefficients or using prior information to **shrink** the parameter space.
   b) Some priors (conjugate) are more useful in terms of analytical expressions and closed-form results. This can hugely reduce the computational burden on sampling algorithms.
   c) Prior distributions should be flexible, meaning that specific information or uncertainty can be easily added.

   With this in mind, there is a literature trying to come up with **structured prior distributions** that are able to (a) shrink the parameter space, (b) imply closed-form expressions for conditional posteriors, and (c) are flexible enough to easily adjust your prior when you change the specification or variables in the model.

To this end, researchers from the University of Minnesota and the Federal Reserve Bank of Minneapolis have proposed a prior which is now known as the **Minnesota Prior** and has become the default for many applications. The idea is to put forward an automatic way to structure the prior for α which is based on just a few hyper-parameters that have the same meaning across any model size or specification.

Consider the following example:

$$\begin{pmatrix} y_t^1 \\ y_t^2 \end{pmatrix} = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} + \begin{pmatrix} A_{11}^1 & A_{12}^1 \\ A_{21}^1 & A_{22}^1 \end{pmatrix} \begin{pmatrix} y_{t-1}^1 \\ y_{t-1}^2 \end{pmatrix} + \begin{pmatrix} A_{11}^2 & A_{12}^2 \\ A_{21}^2 & A_{22}^2 \end{pmatrix} \begin{pmatrix} y_{t-2}^1 \\ y_{t-2}^2 \end{pmatrix} + \begin{pmatrix} u_t^1 \\ u_t^2 \end{pmatrix}$$

Now the Minnesota prior imposes three things:

a) The individual variables y_t^1 and y_t^2 follow a Random Walk. We implement this by setting the prior mean α_0 to zero except for the elements corresponding to A_{11}^1 and A_{22}^1. This is the way we implement shrinkage.
Side-note: You might need to manually adjust this. For instance, growth rates typically show little persistence, so we could also set A_{11}^1 and A_{22}^1 to zero. Alternatively, for level variables with high persistence, we could set A_{11}^1 and A_{22}^1 to high values slightly below 1. Nevertheless, the Random Walk is a good candidate and default in most implementations of the Minnesota prior.

b) The prior covariance matrix V_0 is set to reflect uncertainty about our Random Walk prior mean. That is, we want to express how **certain** we are that (1) all coefficients on lags higher than 1 are zero and (2) coefficients other than on own lags are zero.
We implement this by a set of hyper-parameters that control the tightness of this prior. Formally, this is implemented by specifying the covariance matrix V_0 as a diagonal matrix, where the diagonal elements are set in a structured way. Let V_0^i denote the block of V_0 associated with coefficients in equation i, then the diagonal elements of V_0^i are set according to:

$$V_{0,jj}^i = \begin{pmatrix} \frac{\lambda_1}{l^2} & \text{for coefficients on own lag } l = 1, \dots, p \\ \frac{\lambda_2}{l^2} \frac{\sigma_{ii}}{\sigma_{jj}} & \text{for coefficients on lag } l = 1, \dots, p \text{ of variables } j \neq i \\ \lambda_3 \sigma_{ii} & \text{for coefficients on exogenous variables} \end{pmatrix}$$

This specification implies that as the lag $l = 1, \dots, p$ increases the coefficients are shrunk towards zero. Moreover, by specifying $\lambda_1 < \lambda_2$ we make own lags more likely to be important than lags of other variables. Whereas setting λ_1 close to zero puts greater weight towards the Random Walk assumption. Note that the term $\frac{\sigma_{ii}}{\sigma_{jj}}$ adjusts for differences in the units the variables are measured in. Typically, we set σ_{ii} to the OLS estimate of the standard error of the reduced-form innovations from univariate AR regressions of equation i.

The common practice is then to use the typical natural conjugate priors, i.e. the prior for Σ_u follows an Inverse Wishart prior and the prior for the coefficients $vec(A)$ conditional on Σ_u is normal. Thus, we can make use of the Gibbs sampler by making use of the analytical expressions for the conditional posteriors.

2 Solution to Bayesian Estimation of a VAR model for the US economy including the Zero-Lower-Bound

progs/matlab/BVARZLB_run.m

```
% -----------------------------------------------------------------------
% Run script to do a Bayesian Estimation of a VAR(2) model for US data on
% Federal Funds Rate, Government Bond Yield, Unemployment and Inflation
% from 2007m1 2010m12.
% The prior variance is adjusted to reflect the view that monetary policy
% is at the effective lower bound.
% Results are stored in log files with different names such that one can
% easily use MATLAB's "Compare Selected Files" tool to see differences
% between results.
% -----------------------------------------------------------------------
% Willi Mutschler, January 23, 2024
% willi@mutschler.eu
% -----------------------------------------------------------------------
BVARZLB(0); % no adjustment
BVARZLB(1); % with adjustment for effective lower bound
```

Run the main script by setting the option to false for the first part and to true for the second part. The main script might look like this:

progs/matlab/BVARZLB.m

```
function BVARZLB(prior_adjust_for_ZLB)
% BVARZLB(prior_adjust_for_ZLB)
% -----------------------------------------------------------------------
% Bayesian Estimation of a VAR(2) model for US data on Federal Funds Rate,
% Government Bond Yield, Unemployment and Inflation from 2007m1 2010m12.
% Bayesian estimation with Gibbs Sampling using a Minnesota Prior for the
% VAR coefficients and an Inverse Wishart Prior for the covariance matrix.
% Optionally, the prior variance is adjusted to reflect the view that
% monetary policy is at the effective lower bound (i.e. the federal funds
% rate is unlikely to respond to changes in other variables).
% -----------------------------------------------------------------------
% INPUTS
%       - prior_adjust_for_ZLB : boolean, 1: adjust prior variance to reflect
%                                         the view that  monetary policy is
%                                         at the zero lower bound
% -----------------------------------------------------------------------
% OUTPUTS
% stores results into log files with different names such that one can easily
% use MATLAB's "Compare Selected Files" tool to see differences between results
% -----------------------------------------------------------------------
% Willi Mutschler, January 23, 2024
% willi@mutschler.eu
% -----------------------------------------------------------------------

%% PRELIMINARIES
if nargin < 1
    prior_adjust_for_ZLB = false; % if no input argument was provided
end

```

```matlab
% specification of the VAR model
const = 1;  % 0: no constant, 1: constant, 2: constant and linear trend
p = 2;      % number of lags on dependent variables

% hyper-parameters for Minnesota prior for BVAR model
hyperparams(1) = 0.5; % tightness parameter for Minnesota prior on lags of own variable
hyperparams(2) = 0.5; % tightness parameter for Minnesota prior on lags of other variables
hyperparams(3) = 1;   % tightness parameter for Minnesota prior on exogenous variables (constant, trends, etc)

% settings for Gibbs sampler
nsave = 10000;      % final number of draws to keep
nburn = 30000;      % draws to discard (burn-in)
ntot = nsave+nburn; % total number of draws

%% DATA HANDLING
% load monthly US data on FFR, govt bond yield, unemployment and inflation
USZLB = importdata('../../data/USZLB.csv');
Yraw = USZLB.data;          % Yraw is a matrix with T rows by K columns
[Traw,K] = size(Yraw);      % initial dimensions of dependent variable
Ylag = lagmatrix(Yraw,1:p); % generate lagged Y matrix which will be part of the Z matrix
% define matrix Z which has all the right-hand-side variables and also get rid of NA observations
if const == 0
    Z = transpose(Ylag(p+1:Traw,:));
elseif const == 1
    Z = transpose([ones(Traw-p,1) Ylag(p+1:Traw,:)]);
elseif const == 2
    Z = transpose([ones(Traw-p,1) transpose((p+1):Traw) Ylag(p+1:Traw,:)]);
end
Y = transpose(Yraw(p+1:Traw,:)); % dependent variable in each equation, get rid of NA observations
[totcoeff,T] = size(Z);          % get size of final matrix Z
ZZt = Z*Z';                      % auxiliary matrix product

%% PRIOR SPECIFICATION
% get standard specification of Minnesota Normal-Inverse-Wishard Prior
[alpha_prior, V_prior, inv_V_prior, v_prior, S_prior, inv_S_prior] = BVARMinnesotaPrior(Yraw,const,p,hyperparams);
if prior_adjust_for_ZLB
    % manually adjust for zero-lower bound on nominal interest rates.
    % The interest rate is the first variable and it is reasonable to assume that
    % as monetary policy is constrained at the effective lower bound, other variables
    % do not have an effect on the nominal interest rate. Therefore,  we need to focus
    % on coefficients A1_12, A1_13, A1_14, A2_12, A2_13, A2_14.
    % The prior mean already sets these equal to 0, but we additionally want
    % to use a very small prior variance to reflect the view that we are
    % quite sure that these parameters are very close to zero.

```

```
    % find position of A1_12, A1_13, A1_14, A2_12, A2_13, A2_14
    tmp = zeros(K,K); tmp(1,2:K) = 1;
    Atmp = [zeros(K,1) tmp tmp];
    idx = find(Atmp==1);
    for j = idx
        V_prior(j,j) = 1e-9; % set to small number
    end
    inv_V_prior = diag(1./diag(V_prior));
end
fprintf('prior for coefficient matrix A:\n')
disp(reshape(alpha_prior,K,1+2*K));
fprintf('prior variance for coefficient matrix A (i.e. only diagonal elements of V_prior ordered in same way as coefficient matrix A):\n')
disp(reshape(diag(V_prior),K,1+2*K));

%% GIBBS SAMPLER: INITIALIZATION
A_draws = zeros(K,totcoeff,nsave); % storage for posterior draws of A = [c A_1 A_2]
SIGMAU_draws = zeros(K,K,nsave);   % storage for posterior draws of SIGMAU
% initialize first draw of SIGMAU with OLS values
A_OLS = (Y*Z')/ZZt;                                    % get OLS estimates
resid_OLS = Y - A_OLS*Z;                               % compute OLS residuals
SIGMAU_OLS = (resid_OLS*resid_OLS')./(T-K*p-const); % OLS estimate of error covariance matrix
SIGMAU_j = SIGMAU_OLS;                                 % first draw for Gibbs sampler

%% GIBBS SAMPLER: ALGORITHM
tic; % start timer
waitb = waitbar(0,'Number of iterations'); % open a GUI waitbar
for j = 1:ntot
    if mod(j,1000) == 0
        waitbar(j/ntot); % update waitbar every 1000th step
    end

    % posterior of (alpha|SIGMAU,Y) ~ N(alpha_post,V_post)
    invSIGMAU_j = inv(SIGMAU_j);
    V_post = inv(inv_V_prior + kron(ZZt,invSIGMAU_j));
    alpha_post = V_post*(inv_V_prior*alpha_prior + kron(Z,invSIGMAU_j)*Y(:));
    % check for stability of the VAR coefficients
    is_stable = false;
    while ~is_stable
        V_post = (V_post + V_post.')/2; % make sure V_post is symmetric, i.e. get rid of numerical inefficiencies due to inverses
        alpha_j = mvnrnd(alpha_post,V_post);                    % draw of alpha_j
        A_j = reshape(alpha_j,K,const+K*p);                     % reshape to get A_j
        Acomp = [A_j(:,2:end); eye(K*(p-1)) zeros(K*(p-1),K)]; % companion matrix
        if (max(abs(eig(Acomp)))>1)==0                          % check Eigenvalues of companion matrix
            is_stable = true;                                   % keep stable draw otherwise re-draw
        end
    end

```

```
    % posterior of (SIGMAU|alpha,Y) ~ invWishard(inv(S_post),v_post)
    v_post = T + v_prior;
    S_post = S_prior + (Y - A_j*Z)*transpose(Y - A_j*Z);
    SIGMAU_j = inv(wishrnd(inv(S_post),v_post));

    % store results if burn-in phase is passed
    if j > nburn
        A_draws(:,:,j-nburn) = A_j;
        SIGMAU_draws(:,:,j-nburn) = SIGMAU_j;
    end
end
close(waitb); % close GUI waitbar
toc; % stop and display timer

%% INFERENCE ON POSTERIOR DRAWS AND COMPARISON WITH OLS
VAR_OLS = VARReducedForm(Yraw,p);
[VAR_OLS.eq1.beta VAR_OLS.eq2.beta VAR_OLS.eq3.beta VAR_OLS.eq4.beta]'
if prior_adjust_for_ZLB
    diary('BVARZLB_results_withZLB.log'); % open log file to save results into a text file
else
    diary('BVARZLB_results_noZLB.log');
end
fprintf('OLS estimate of A:\n')
A_OLS
fprintf('OLS estimate of SIGMAU:\n')
SIGMAU_OLS

fprintf('Posterior mean of A:\n')
A_mean     = mean(A_draws,3)
fprintf('Posterior mean of SIGMAU:\n')
SIGMA_mean = mean(SIGMAU_draws,3)

fprintf('OLS standard error of A:\n')
[VAR_OLS.eq1.bstd VAR_OLS.eq2.bstd VAR_OLS.eq3.bstd VAR_OLS.eq4.bstd]'
fprintf('Posterior standard deviation of A:\n')
se_A      = std(A_draws,0,3)

fprintf('OLS lower 5th confidence interval of A:\n')
[VAR_OLS.eq1.bint(:,1) VAR_OLS.eq2.bint(:,1) VAR_OLS.eq3.bint(:,1) VAR_OLS.eq4.bint(:,1)]'
fprintf('Posterior lower 5th percentile of A:\n')
LOWER_A = prctile(A_draws,5,3)

fprintf('OLS upper 95th confidence interval of A:\n')
[VAR_OLS.eq1.bint(:,2) VAR_OLS.eq2.bint(:,2) VAR_OLS.eq3.bint(:,2) VAR_OLS.eq4.bint(:,2)]'
fprintf('Posterior upper 95th percentile of A:\n')
UPPER_A = prctile(A_draws,95,3)

diary off; % close log file
```