clc

clear all

%load('C:\Users\islamr\Dropbox\Mardi Meetings\Meeting 10\Phase16\Raisul2.mat');

load('C:\Users\Mihan\Dropbox\Mardi Meetings\Meeting 14\Raisul2.mat');

CP1=Raisul2;

%CPNew=CP1;

CP1(:,21:24)=[]; % remove 13th country

%CP1(:,22:23)=[]; % remove 13th country

%Remember when u omit number 1 column, number 2 becomes number 1. Remember

%it

CPNew = log(CP1);

CPNew= 100\*(trimr(CPNew,1,0)-trimr(CPNew,0,1));

window\_Size = 2;

delta = (1/window\_Size)\*ones(1,window\_Size);

gama=1;

filt=filter(delta,gama,CPNew);

[nRow, nCol] = size(filt);

windowSize=100;

% creating empty array to store results

contToOthers=[];

contFromOthers=[];

%for J = 1:windowSize: (nRow-windowSize) % size by size

for J = 1:(nRow-windowSize) % continued window

wend=J+windowSize-1;

y = filt(J:wend,:);

z = bsxfun(@rdivide,bsxfun(@minus,y,mean(y)),std(y));

%disp(' Mean Median Max Min Std.dev. Skew Kurt')

%disp( [ mean(y)' median(y)' max(y)' min(y)' std(y)' mean(z.^3)' mean(z.^4)'] )

x = [ones(length(y)-2,1) trimr(y,1,1) trimr(y,0,2) ];

b = x\trimr(y,2,0);

%It is simple annual growth rate in chain multiplication format

% We have 3 columns of data which are the growth rates only...We are

% calculating the growth rate by simply omitting the first and last 12

% data which is 12 months (annual) data and than by

% deductions we are calculating the annual rate of growth.

%for example tmp = 100\*(trimr(yvar(:,2:4),12,0) - trimr(yvar(:,2:4),0,12));

%for the interest rates we are only fixing the dimension as no growth

%rate needs calculating.

% for example y = [ trimr(yvar(:,1),12,0) tmp];

% t = length(y);

% We now Obtain estimates of the VAR(2) by OLS to each equation

% Now, the trimr portions are only doing iteration by number of VAR,this process sets restrictions on the system..

% here upto 2 lags as we are calculating VAR(2). We are using ones only

% for matrix concatenation...

%bols = [ones(t-2,1) trimr(y,1,1) trimr(y,0,2)]\trimr(y,2,0);

%Standard OLS

%a1 = x\y;

%u1 = y - x\*a1;

% first line is calculating dimension of annual percentage growt rate and than OLS

%% Taking total number of rows and columns

[rcnt, ccnt] = size(b);

%% need to fix values for these variables 16, 8, 9

% mu = trimr(b,0,rcnt-1)'; % Vector of intercepts

phi1 = trimr(b,1,ccnt)'; % Lag 1 parameter estimates

phi2 = trimr(b,ccnt+1,0)'; % Lag 2 parameter estimates

%% Generate VMA (non-orthogonalized) for horizons 1 to 10

% Recursive estimation: Transforming VAR to VMA using recursion.

si1 = eye(size(y,2));

si2 = phi1;

si3 = phi1\*si2 + phi2;

si4 = phi1\*si3 + phi2\*si2;

si5 = phi1\*si4 + phi2\*si3;

si6 = phi1\*si5 + phi2\*si4;

si7 = phi1\*si6 + phi2\*si5;

si8 = phi1\*si7 + phi2\*si6;

si9 = phi1\*si8 + phi2\*si7;

si10 = phi1\*si9 + phi2\*si8;

% can we use for loop

% Generate VMA (orthogonalized) for horizons 1 to 10

% v is the disturbance vector at time that is used to estimate vc. vc

% is the residual covariance matrix.d identifies the diagonal vector

% where the structura shocks are located (own volaitility shocks). s is

% the cholesky decomposition used to acheiev a lower diagonal matrix.

v = trimr(y,2,0) - x\*b; % VAR residuals

vc = v'\*v/length(v);

d = diag(vc);

s = chol(vc);

% now we calculate the impulse response Ir (page 498) from cholesky

% decompostion

ir1 = si1\*s;

ir2 = si2\*s;

ir3 = si3\*s;

ir4 = si4\*s;

ir5 = si5\*s;

ir6 = si6\*s;

ir7 = si7\*s;

ir8 = si8\*s;

ir9 = si9\*s;

ir10 = si10\*s;

% Compute variance decompositions for horizons 1 to 10

vd1 = ir1.^2;

vd1 = 100\*bsxfun(@rdivide,vd1,sum(vd1,2));

vd2 = ir1.^2 + ir2.^2;

vd2 = 100\*bsxfun(@rdivide,vd2,sum(vd2,2));

vd3 = ir1.^2 + ir2.^2 + ir3.^2;

vd3 = 100\*bsxfun(@rdivide,vd3,sum(vd3'));

vd4 = ir1.^2 + ir2.^2 + ir3.^2 + ir4.^2;

vd4 = 100\*bsxfun(@rdivide,vd4,sum(vd4'));

vd5 = ir1.^2 + ir2.^2 + ir3.^2 + ir4.^2 + ir5.^2;

vd5 = 100\*bsxfun(@rdivide,vd5,sum(vd5'));

vd6 = ir1.^2 + ir2.^2 + ir3.^2 + ir4.^2 + ir5.^2 + ir6.^2;

vd6 = 100\*bsxfun(@rdivide,vd6,sum(vd6'));

vd7 = ir1.^2 + ir2.^2 + ir3.^2 + ir4.^2 + ir5.^2 + ir6.^2 + ir7.^2;

vd7 = 100\*bsxfun(@rdivide,vd7,sum(vd7'));

vd8 = ir1.^2 + ir2.^2 + ir3.^2 + ir4.^2 + ir5.^2 + ir6.^2 + ir7.^2 + ir8.^2;

vd8 = 100\*bsxfun(@rdivide,vd8,sum(vd8'));

vd9 = ir1.^2 + ir2.^2 + ir3.^2 + ir4.^2 + ir5.^2 + ir6.^2 + ir7.^2 + ir8.^2 + ir9.^2;

vd9 = 100\*bsxfun(@rdivide,vd9,sum(vd9'));

vd10 = ir1.^2 + ir2.^2 + ir3.^2 + ir4.^2 + ir5.^2 + ir6.^2 + ir7.^2 + ir8.^2 + ir9.^2 + ir10.^2;

vd10 = 100\*bsxfun(@rdivide,vd10,sum(vd10,2));

%% Change the country list

str = [ 'US ' ;

'AUS ' ;

'INDIA ' ;

'JAP ' ;

'MALAY ' ;

'NZ ' ;

'SINGAPR ' ;

'PHILLI ' ;

'SOUTHKOR ' ;

'SriLanka ' ;

'THAILAND ' ;

'Nigeria ' ;

'Venezuela ' ;

'Equador ' ;

'Kuwait ' ;

'Iraq ' ;

'Saudi ' ;

'China ' ;

'Israel ' ;

'Canada ' ] ;

% 'Commodity ' ];

%disp('Variance decomposition at period 10')

%disp(vd10);

tmp=sum(vd10,2)-diag(vd10);

disp(J);

disp('Contribution From Others')

disp( [str num2str(tmp) ]);

contFromOthers = [contFromOthers, tmp];

tmp2=(sum(vd10)-diag(vd10)')';

%contFromOthers{J}=(tmp2);

disp('Contribution To Others')

disp( [str num2str(tmp2) ]);

contToOthers = [contToOthers, tmp2];

end

dlmwrite('contFromOthers.csv', contFromOthers);

dlmwrite('contToOthers.csv', contToOthers);

xvals=1:3692;

%When we do suppose, 2 week forecast horizon, in that case, it will be

%xvals=1:(number of rows)...this is found in excel files....and this can be

%customizable at any point to point.......

plot(xvals,contToOthers(20,:), xvals, contFromOthers(20,:))

%plot(xvals,contToOthers, xvals, contFromOthers)

clc

clear all

%load('C:\Users\islamr\Dropbox\Mardi Meetings\Meeting 10\Phase16\Raisul2.mat');

load('C:\Users\Mihan\Dropbox\Mardi Meetings\Meeting 14\Raisul2.mat');

CPnew=Raisul2;

CPnew(:,21)=[]; % remove 13th country

CPnew(:,22:23)=[]; % remove 13th country

%CPnew(:,10)=[]; % remove 1st country

%CPnew(:,5)=[]; % remove 3rd country

contToOthers=[];

contFromOthers=[];

y = log(CPnew);

y= 100\*(trimr(y,1,0)-trimr(y,0,1));

z = bsxfun(@rdivide,bsxfun(@minus,y,mean(y)),std(y));

x= linspace(1,3792,3792);

yp = polyfit(x',y(:,1),4);

yp1=polyval(yp,x);

resid1 = y(:,1)- yp1';

x= linspace(1,3792,3792);

ypl = polyfit(x',y(:,21),4); % caution notice carefully see cftool

yp21=polyval(ypl,x);

resid21 = y(:,21)- yp21';

xx=crosscorr(resid1,resid21)

%fitnessIntegralMethod=100\*(1-norm((y-yp1)/norm((y)-mean(y)))

%%

disp(' Mean Median Max Min Std.dev. Skew Kurt')

disp( [ mean(y)' median(y)' max(y)' min(y)' std(y)' mean(z.^3)' mean(z.^4)'] )

x = [ones(length(y)-2,1) trimr(y,1,1) trimr(y,0,2) ];

b = x\trimr(y,2,0);

%% Taking total number of rows and columns

[rcnt, ccnt] = size(b)

%% need to fix values for these variables 16, 8, 9

% mu = trimr(b,0,rcnt-1)'; % Vector of intercepts

phi1 = trimr(b,1,ccnt)'; % Lag 1 parameter estimates

phi2 = trimr(b,ccnt+1,0)'; % Lag 2 parameter estimates

%% Generate VMA (non-orthogonalized) for horizons 1 to 10

si1 = eye(size(y,2));

si2 = phi1;

si3 = phi1\*si2 + phi2;

si4 = phi1\*si3 + phi2\*si2;

si5 = phi1\*si4 + phi2\*si3;

si6 = phi1\*si5 + phi2\*si4;

si7 = phi1\*si6 + phi2\*si5;

si8 = phi1\*si7 + phi2\*si6;

si9 = phi1\*si8 + phi2\*si7;

si10 = phi1\*si9 + phi2\*si8;

% Generate VMA (orthogonalized) for horizons 1 to 10

v = trimr(y,2,0) - x\*b; % VAR residuals

vc = v'\*v/length(v);

d = diag(vc);

s = chol(vc);

ir1 = si1\*s;

ir2 = si2\*s;

ir3 = si3\*s;

ir4 = si4\*s;

ir5 = si5\*s;

ir6 = si6\*s;

ir7 = si7\*s;

ir8 = si8\*s;

ir9 = si9\*s;

ir10 = si10\*s;

% Compute variance decompositions for horizons 1 to 10

vd1 = ir1.^2;

vd1 = 100\*bsxfun(@rdivide,vd1,sum(vd1,2));

vd2 = ir1.^2 + ir2.^2;

vd2 = 100\*bsxfun(@rdivide,vd2,sum(vd2,2));

vd3 = ir1.^2 + ir2.^2 + ir3.^2;

vd3 = 100\*bsxfun(@rdivide,vd3,sum(vd3'));

vd4 = ir1.^2 + ir2.^2 + ir3.^2 + ir4.^2;

vd4 = 100\*bsxfun(@rdivide,vd4,sum(vd4'));

vd5 = ir1.^2 + ir2.^2 + ir3.^2 + ir4.^2 + ir5.^2;

vd5 = 100\*bsxfun(@rdivide,vd5,sum(vd5'));

vd6 = ir1.^2 + ir2.^2 + ir3.^2 + ir4.^2 + ir5.^2 + ir6.^2;

vd6 = 100\*bsxfun(@rdivide,vd6,sum(vd6'));

vd7 = ir1.^2 + ir2.^2 + ir3.^2 + ir4.^2 + ir5.^2 + ir6.^2 + ir7.^2;

vd7 = 100\*bsxfun(@rdivide,vd7,sum(vd7'));

vd8 = ir1.^2 + ir2.^2 + ir3.^2 + ir4.^2 + ir5.^2 + ir6.^2 + ir7.^2 + ir8.^2;

vd8 = 100\*bsxfun(@rdivide,vd8,sum(vd8'));

vd9 = ir1.^2 + ir2.^2 + ir3.^2 + ir4.^2 + ir5.^2 + ir6.^2 + ir7.^2 + ir8.^2 + ir9.^2;

vd9 = 100\*bsxfun(@rdivide,vd9,sum(vd9'));

vd10 = ir1.^2 + ir2.^2 + ir3.^2 + ir4.^2 + ir5.^2 + ir6.^2 + ir7.^2 + ir8.^2 + ir9.^2 + ir10.^2;

vd10 = 100\*bsxfun(@rdivide,vd10,sum(vd10,2));

%% Change the country list

str = [ 'US ' ;

'AUS ' ;

'INDIA ' ;

'JAP ' ;

'MALAY ' ;

'NZ ' ;

'SINGAPR ' ;

'PHILLI ' ;

'SOUTHKOR ' ;

'SriLanka ' ;

'THAILAND ' ;

'Nigeria ' ;

'Venezuela ' ;

'Equador ' ;

'Kuwait ' ;

'Iraq ' ;

'Saudi ' ;

'China ' ;

'Israel ' ;

'Canada ' ;

'Commodity ' ];

disp('Variance decomposition at period 10')

disp(vd10);

tmp=sum(vd10,2)-diag(vd10);

disp('Contribution From Others')

disp( [str num2str(tmp) ]);

contFromOthers = [contFromOthers, tmp];

tmp2=(sum(vd10)-diag(vd10)')';

disp('Contribution To Others')

disp( [str num2str(tmp2) ]);

contToOthers = [contToOthers, tmp2];

dlmwrite('contFromOthers.csv', contFromOthers);

dlmwrite('contToOthers.csv', contToOthers);