Vector Autoregressive (VARs) Models

Reduced form models can be stated as:

$$Y_t = \Delta_0 + \Delta_1 Y_{t-1} + \in_t$$

$$\begin{pmatrix} x_t \\ y_t \end{pmatrix} = \begin{pmatrix} \delta_{10} \\ \delta_{20} \end{pmatrix} + \begin{pmatrix} \delta_{11} & \delta_{12} \\ \delta_{21} & \delta_{22} \end{pmatrix} \begin{pmatrix} x_{t-1} \\ y_{t-1} \end{pmatrix} + \begin{pmatrix} e_{1t} \\ e_{2t} \end{pmatrix}$$

where: $\in_t = A^{-1}\varepsilon_t$

$$A^{-1}\Gamma_0 = \Delta_0 = \begin{pmatrix} \delta_{10} \\ \delta_{20} \end{pmatrix} = \begin{pmatrix} 0.1 \\ 0.1 \end{pmatrix}$$

$$A^{-1}\Gamma_1 = \Delta_1 = \begin{pmatrix} \delta_{11} & \delta_{12} \\ \delta_{21} & \delta_{22} \end{pmatrix} = \begin{pmatrix} 0.4 & 0.1 \\ 0.2 & 0.5 \end{pmatrix}$$

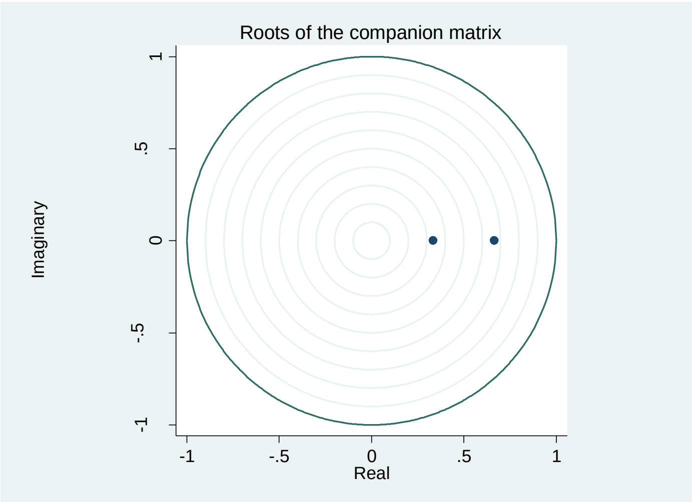
then, variance-covariance matrix of $\in_t$ is $A^{-1}\Sigma A^{-1\prime} = \Omega = \begin{pmatrix} \sigma^2_{e_1} & \sigma^2_{e_1 e_2} \\ \sigma^2_{e_2 e_1} & \sigma^2_{e_2} \end{pmatrix} = \begin{pmatrix} 16 & -4 \\ -4 & 25 \end{pmatrix}$

Do file command in generating VARs system:

```
set obs 500
g t=_n
tsset t
matrix c=(1, -0.2\ -0.2, 1)
corr2data u1 u2, n(500) mean(0 0) sds(4 5) corr(c)
g x=0.1 in 1
g y=0.1 in 1
forvalues i = 2(1)500 {
replace x=0.1+0.4*l.x+0.1*l.y+u1 in `i'
replace y=0.1+0.2*l.x+0.5*l.y+u2 in `i'
}
```

```
. varsoc x y, maxlag(5)

   Selection-order criteria
   Sample:  6 - 500                             Number of obs      =       495
  +---------------------------------------------------------------------------+
  |lag |    LL      LR      df    p      FPE       AIC      HQIC      SBIC    |
  |----+----------------------------------------------------------------------|
  |  0 | -3046.22                      765.308    12.316   12.3227    12.333  |
  |  1 | -2871.01  350.42*   4  0.000   383.19*  11.6243*  11.6443*  11.6752* |
  |  2 | -2869.73  2.5675    4  0.633  387.419   11.6353   11.6686   11.7202  |
  |  3 |  -2866.6  6.2625    4  0.180  388.782   11.6388   11.6855   11.7577  |
  |  4 |  -2864.2  4.7814    4  0.310   391.32   11.6453   11.7053   11.7982  |
  |  5 | -2862.63  3.1391    4  0.535  395.184   11.6551   11.7284    11.842  |
  +---------------------------------------------------------------------------+
   Endogenous:  x y
    Exogenous:  _cons

. var x y, lag(1/1)

Vector autoregression

Sample:  2 - 500                                   No. of obs      =       499
Log likelihood = -2895.765                         AIC             =  11.63032
FPE            =  385.5104                         HQIC            =   11.6502
Det(Sigma_ml)  =  376.3501                         SBIC            =  11.68097
```

```
Equation           Parms      RMSE     R-sq      chi2     P>chi2
----------------------------------------------------------------
x                    3     3.99618   0.1943    120.35   0.0000
y                    3     4.98323   0.3635  285.0113   0.0000
----------------------------------------------------------------

------------------------------------------------------------------------------
             |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
x            |
           x |
         L1. |    .425351   .0402202    10.58   0.000     .3465209     .504181
             |
           y |
         L1. |   .0827235   .0286777     2.88   0.004     .0265162    .1389309
             |
       _cons |    .113024   .1787015     0.63   0.527    -.2372246    .4632726
-------------+----------------------------------------------------------------
y            |
           x |
         L1. |   .2657108   .0501544     5.30   0.000       .16741    .3640117
             |
           y |
         L1. |   .5726519    .035761    16.01   0.000     .5025616    .6427423
             |
       _cons |   .0608357   .2228402     0.27   0.785    -.3759231    .4975945
------------------------------------------------------------------------------

. varstable, graph

   Eigenvalue stability condition
  +----------------------------------------+
  |        Eigenvalue         |   Modulus  |
  |--------------------------+-------------|
  |   .6645458                |   .664546  |
  |   .3334571                |   .333457  |
  +----------------------------------------+
   All the eigenvalues lie inside the unit circle.
   VAR satisfies stability condition.
```




```
. *Proof eigenvalues

. mat Dr=e(b)

. mat list Dr

Dr[1,6]
```

```
            x:          x:          x:          y:          y:          y:
            L.          L.                      L.          L.
            x           y       _cons           x           y       _cons
y1  .42535096  .08272355    .113024  .26571084  .57265191  .06083569

. mat D1=(el(Dr,1,1), el(Dr,1,2)\el(Dr,1,4), el(Dr,1,5))

. mat list D1

D1[2,2]
           c1          c2
r1  .42535096  .08272355
r2  .26571084  .57265191

. mat eigenvalues L C=D1

. mat list L

L[1,2]
             c1          c2
real  .33345707   .6645458

. vargranger

   Granger causality Wald tests
  +------------------------------------------------------------------+
  |          Equation           Excluded |   chi2     df Prob > chi2 |
  |--------------------------------------+---------------------------|
  |                 x                  y |  8.3209     1    0.004    |
  |                 x                ALL |  8.3209     1    0.004    |
  |--------------------------------------+---------------------------|
  |                 y                  x |  28.067     1    0.000    |
  |                 y                ALL |  28.067     1    0.000    |
  +------------------------------------------------------------------+

. *Impulse Response Function (IRF) Analysis

. mat list D1

D1[2,2]
           c1          c2
r1  .42535096  .08272355
r2  .26571084  .57265191

. mat IRF0=(1\0)

. mat list IRF0

IRF0[2,1]
    c1
r1   1
r2   0

. mat IRF1=D1*IRF0

. mat list IRF1

IRF1[2,1]
           c1
r1  .42535096
r2  .26571084

. mat IRF2=D1*IRF1

. mat list IRF2

IRF2[2,1]
```

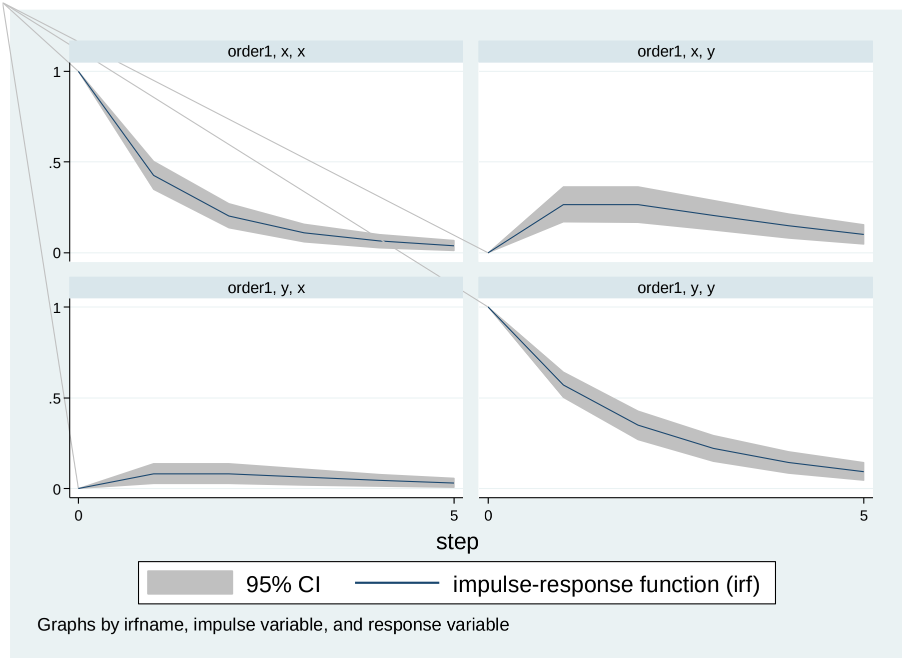
```
           c1
r1  .20290398
r2  .26518018

. mat IRF3=D1*IRF2

. mat list IRF3

IRF3[2,1]
           c1
r1  .10824205
r2  .20576972

. mat IRF4=D1*IRF3

. mat list IRF4

IRF4[2,1]
           c1
r1  .06306286
r2  .14659551

. mat IRF5=D1*IRF4

. mat list IRF5

IRF5[2,1]
           c1
r1  .03895075
r2  .10070468

. mat IRF=(IRF0'\IRF1'\IRF2'\IRF3'\IRF4'\IRF5')

. mat rownames IRF = IRF0 IRF1 IRF2 IRF3 IRF4 IRF5

. mat colnames IRF = x y

. mat list IRF

IRF[6,2]
              x          y
IRF0          1          0
IRF1  .42535096  .26571084
IRF2  .20290398  .26518018
IRF3  .10824205  .20576972
IRF4  .06306286  .14659551
IRF5  .03895075  .10070468

. irf create order1, order(x y) step(5) set(irf01)
(file irf01.irf created)
(file irf01.irf now active)
(file irf01.irf updated)

. irf table irf, impulse(x y) response(x y)
```

Results from order1

step	(1) irf	(1) Lower	(1) Upper	(2) irf	(2) Lower	(2) Upper
0	1	1	1	0	0	0
1	.425351	.346521	.504181	.265711	.16741	.364012
2	.202904	.135348	.27046	.26518	.167287	.363073
3	.108242	.058281	.158203	.20577	.122689	.28885
4	.063063	.026316	.09981	.146596	.079092	.214099
5	.038951	.011828	.066073	.100705	.047266	.154144

step	(3) irf	(3) Lower	(3) Upper	(4) irf	(4) Lower	(4) Upper
0	0	0	0	1	1	1
1	.082724	.026516	.138931	.572652	.502562	.642742
2	.082558	.026958	.138159	.349911	.270834	.428987
3	.064062	.019269	.108855	.222314	.150064	.294564
4	.045639	.011503	.079776	.14433	.083219	.205441
5	.031352	.005879	.056825	.094778	.045313	.144242

```
95% lower and upper bounds reported
(1) irfname = order1, impulse = x, and response = x
(2) irfname = order1, impulse = x, and response = y
(3) irfname = order1, impulse = y, and response = x
(4) irfname = order1, impulse = y, and response = y


. irf graph irf, impulse(x y) response(x y)
```




```
. *Orthogonal Impulse Response Function Analysis (OIRF)

. mat W=e(Sigma)

. mat list W

symmetric W[2,2]
            x          y
x   15.873479
y  -3.9318246   24.68327

. mat P_1=cholesky(W)

. mat list P_1

P_1[2,2]
            x          y
x   3.9841534          0
y  -.98686576  4.8692264

. mat U=(1\0)
```

```
. mat OIRF0=P_1*U

. mat list OIRF0

OIRF0[2,1]
           c1
x   3.9841534
y  -.98686576

. mat OIRF1=D1*OIRF0

. mat list OIRF1

OIRF1[2,1]
           c1
r1  1.6130264
r2   .4935022

. mat OIRF2=D1*OIRF1

. mat list OIRF2

OIRF2[2,1]
           c1
r1  .72692659
r2  .71120359

. mat OIRF3=D1*OIRF2

. mat list OIRF3

OIRF3[2,1]
           c1
r1   .3680322
r2  .60042437

. mat OIRF4=D1*OIRF3

. mat list OIRF4

OIRF4[2,1]
           c1
r1  .20621208
r2  .44162431

. mat OIRF5=D1*OIRF4

. mat list OIRF5

OIRF5[2,1]
           c1
r1  .12424524
r2  .30768979

. mat OIRF=(OIRF0'\OIRF1'\OIRF2'\OIRF3'\OIRF4'\OIRF5')

. mat rownames OIRF=OIRF0 OIRF1 OIRF2 OIRF3 OIRF4 OIRF5

. mat colnames OIRF=x y
```

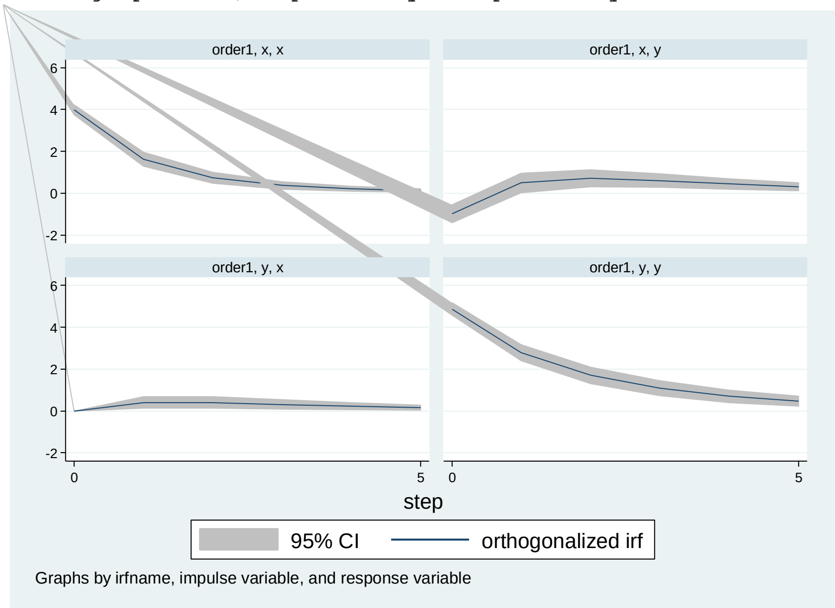
```
. mat list OIRF

OIRF[6,2]
                 x           y
OIRF0   3.9841534  -.98686576
OIRF1   1.6130264    .4935022
OIRF2   .72692659   .71120359
OIRF3    .3680322   .60042437
OIRF4   .20621208   .44162431
OIRF5   .12424524   .30768979

. irf table oirf, impulse(x y) response(x y)
```

Results from order1

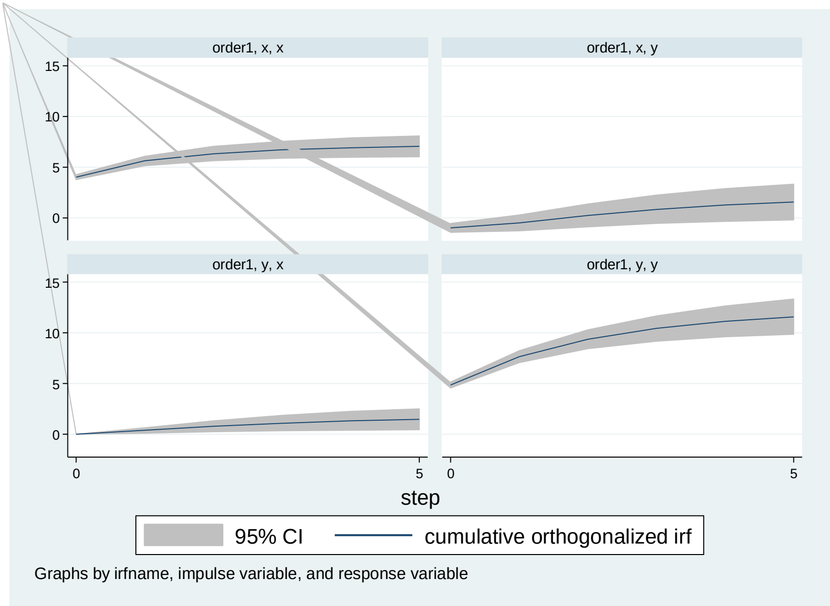
step	(1) oirf	(1) Lower	(1) Upper	(2) oirf	(2) Lower	(2) Upper
0	3.98415	3.73697	4.23134	-.986866	-1.41846	-.555275
1	1.61303	1.27674	1.94931	.493502	.025392	.961613
2	.726927	.464225	.989628	.711204	.304563	1.11784
3	.368032	.187769	.548295	.600424	.281124	.919725
4	.206212	.080608	.331817	.441624	.196456	.686793
5	.124245	.034493	.213998	.30769	.121523	.493856

step	(3) oirf	(3) Lower	(3) Upper	(4) oirf	(4) Lower	(4) Upper
0	0	0	0	4.86923	4.56713	5.17132
1	.4028	.127975	.677624	2.78837	2.40574	3.171
2	.401995	.130118	.673873	1.70379	1.30451	2.10308
3	.311933	.092968	.530898	1.0825	.72434	1.44065
4	.222229	.055441	.389017	.702777	.402037	1.00352
5	.152661	.028267	.277056	.461495	.218946	.704045

```
95% lower and upper bounds reported
(1) irfname = order1, impulse = x, and response = x
(2) irfname = order1, impulse = x, and response = y
(3) irfname = order1, impulse = y, and response = x
(4) irfname = order1, impulse = y, and response = y

. irf graph oirf, impulse(x y) response(x y)
```



```
. irf graph coirf, impulse(x y) response(x y)
```




```
. sca Pc11=el(P_1,1,1)

. sca Pc12=el(P_1,1,2)

. sca Wc11=el(W,1,1)

. sca wdx0=(Pc11^2)/(Wc11)

. sca wdy0=(Pc12^2)/(Wc11)

. sca list wdx0 wdy0
      wdx0 =          1
      wdy0 =          0

. mat D1P=D1*P_1

. mat list D1P

D1P[2,2]
            x          y
r1  1.6130264  .40279968
r2   .4935022  2.7883718

. mat D1PT=D1P*D1P'

. mat list D1PT

symmetric D1PT[2,2]
           r1         r2
r1  2.7641019
r2  1.9191874  8.0185616

. sca D1Pc11=el(D1P,1,1)

. sca D1Pc12=el(D1P,1,2)

. sca D1PTc11=el(D1PT,1,1)

. sca wdx1=(D1Pc11^2+Pc11^2)/(D1PTc11+Wc11)

. sca wdy1=(D1Pc12^2+Pc12^2)/(D1PTc11+Wc11)
```

```
. sca list wdx1 wdy1
      wdx1 =   .9912946
      wdy1 =   .0087054

. mat D2P=D1*D1*P_1

. mat list D2P

D2P[2,2]
            x          y
r1  .72692659  .40199523
r2  .71120359  1.7037947

. mat D2PT=D2P*D2P'

. mat list D2PT

symmetric D2PT[2,2]
           r1         r2
r1  .69002243
r2  1.2019101  3.4087268

. sca D2Pc11=el(D2P,1,1)

. sca D2Pc12=el(D2P,1,2)

. sca D2PTc11=el(D2PT,1,1)

. sca wdx2=(D2Pc11^2+D1Pc11^2+Pc11^2)/(D2PTc11+D1PTc11+Wc11)

. sca wdy2=(D2Pc12^2+D1Pc12^2+Pc12^2)/(D2PTc11+D1PTc11+Wc11)

. sca list wdx2 wdy2
      wdx2 =  .98324429
      wdy2 =  .01675571

. mat FEVD=(wdx0, wdy0 \ wdx1, wdy1 \ wdx2, wdy2)

. mat rownames FEVD = t1 t2 t3

. mat colnames FEVD = x y

. mat list FEVD

FEVD[3,2]
            x          y
t1          1          0
t2   .9912946   .0087054
t3  .98324429  .01675571

. irf table fevd, impulse(x y) response(x)

                                Results from order1
+----------------------------------------------------------------------------------+
|        |    (1)          (1)          (1)      |    (2)          (2)          (2)     |
|  step  |   fevd        Lower        Upper      |   fevd        Lower        Upper     |
|--------+------------------------------------+------------------------------------|
|0       | 0            0            0          | 0            0            0          |
|1       | 1            1            1          | 0            0            0          |
|2       | .991295      .979376      1.00321    | .008705      -.003213     .020624    |
|3       | .983244      .960721      1.00577    | .016756      -.005767     .039279    |
|4       | .978469      .949637      1.0073     | .021531      -.007302     .050363    |
|5       | .976057      .94391       1.0082     | .023943      -.008204     .05609     |
+----------------------------------------------------------------------------------+
95% lower and upper bounds reported
(1) irfname = order1, impulse = x, and response = x
```

(2) irfname = order1, impulse = y, and response = x

. irf table fevd, impulse(x y) response(y)

Results from order1

step	(1) fevd	(1) Lower	(1) Upper	(2) fevd	(2) Lower	(2) Upper
0	0	0	0	0	0	0
1	.039456	.005975	.072937	.960544	.927063	.994025
2	.037229	.014853	.059605	.962771	.940395	.985147
3	.047722	.01977	.075673	.952278	.924327	.98023
4	.055356	.021366	.089347	.944644	.910653	.978634
5	.059449	.022057	.096841	.940551	.903159	.977943

95% lower and upper bounds reported
(1) irfname = order1, impulse = x, and response = y
(2) irfname = order1, impulse = y, and response = y