

### 3.20 VAR モデル(2)

ver.0.1

引き続き VAR の枠組みの中での分析を続けます。変数間の因果関係や、その変数を VAR の中に含めるか否かをテストする方法を見ていきます。

#### Granger Causality Test

このテストは、VAR モデルの推定式のうちの 1 本を選んで、そのなかのある変数のラグをすべて除いた制限モデルともとの非制限モデルの RSS を用いて、F 分布によるテストで、それらの取り除いた変数のもとの変数が従属変数に因果性を持っているかどうかを調べるものです。すなわち、例えば

$$\begin{aligned} Y_t &= \alpha_{11} + \alpha_{12} Y_{t-1} + \alpha_{13} L_{t-1} + \alpha_{14} K_{t-1} + \alpha_{15} Y_{t-2} + \alpha_{16} L_{t-2} + \alpha_{17} K_{t-2} + u_{1t} \\ Y_t &= \alpha_{11} + \alpha_{12} Y_{t-1} + \alpha_{14} K_{t-1} + \alpha_{15} Y_{t-2} + \alpha_{17} K_{t-2} + u_{1r} \end{aligned}$$

の非制限モデルと制限モデルを考えて、それぞれの残差平方和  $RSS_u$  と  $RSS_r$  を用いて、

$$F = \frac{(RSS_r - RSS_u) / 2}{RSS_u / (n - 7)} \sim F(2, n - 7)$$

でテストを行なうものです。ここでは  $\alpha_{13}$  と  $\alpha_{16}$  が 0 に制限されていますから自由度は分子が 2、分母はもとのパラメータの数の 7 がきます。ここでテストする仮説は、

$$H_0: \alpha_{13} = \alpha_{16} = 0$$

$H_1: H_0$  is NOT true. ( L から Y に Granger の意味での因果性がある )

となります。F がカットオフ値より大きく  $H_0$  を棄却できるなら、Granger の意味での因果性は、取り除いた変数 L から従属変数 Y に対して、存在することになります。これを、異なる変数の組み合わせ回数だけ繰り返します。

#### プログラム

```
new; cls;
load data[40,3]=d:datafile13.txt;
Y=data[2:40,1]; L=data[2:40,2]; K=data[2:40,3];
data=Y~L~K;
call granger(data,2,0.05);

proc(0)=granger(data,lag,pp);
local k,c,d;
k=cols(data);
c=1;
```

```

do while c<=k;
    d=1;
    do while d<=k;
        if c/=d;
            print "          =====";
            print/rz c "<-";;print/lz d;
            print "          =====";
            call grangermain(data,lag,c,d,pp);
        endif;
        d=d+1;
    endo;
    c=c+1;
endo;
endp;

proc(0)=grangermain(data,lag,c,d,pp);
    local n,k,n1,i,xlag,y,x,k1,yc,b,e,rssu,labelx,br,er,rssr,f,p;
    if c==d;
        retp;
    endif;
    n=rows(data); k=cols(data); n1=n-lag;
    i=2; xlag=data[lag:n-1,.];
    do while i<=lag;
        xlag=xlag~data[lag+1-i:n-i,.];
        i=i+1;
    endo;
    y=data[lag+1:n,.];
    x=ones(n-lag,1)~xlag;
    k1=cols(x);
    yc=y[.,c];
    b=inv(x'x)*x'yc;
    e=yc-x*b;
    rssu=e'e;
/* Restricted Model */
    i=1; labelx=0;
    do while i<=lag;

```

```

        labelx=labelx~seqa(1,1,k)';
        i=i+1;
    endo;
    x=labelx | x;
    x=x';
    x=delif(x,(x[.,1] .== d));
    x=x'; /* Try to see print x; */
    x=trimr(x,1,0);
    yc=y[.,c];
    br=inv(x'x)*x'yc;
    er=yc-x*br;
    rssr=er' er;
/* Calculate F */
    f=((rssr-rssu)/lag) / (rssu/(n1-k1));
    print/rz "F(" lag ", " n1-k1 ") =";;print f;;
    p=cdfc(f,lag,n1-k1); print/lo ":P-value=" p;
    print/lz pp*100 "% significance level      ";;
    if p<pp;
        print "Reject H0[ Granger's causality exists!] ";
    else;
        print "NOT reject H0";
    endif;
    print;
    retp;
endp;

```

ここでは少しテクニックを使っています。これまでどおりのVARのプログラムのうち、従属変数  $y$  の行列の  $c$  列目だけを従属変数にします。これにより 1 本だけの推定をし、それにもとづく非制限モデルの RSS をまず求めます。次に、このプログラム方法の場合、従属変数は、 $1, Y_{t-1}, L_{t-1}, K_{t-1}, Y_{t-2}, L_{t-2}, K_{t-2}$  の順になっていますから、

```

i=1; labelx=0;
do while i<=lag;
    labelx=labelx~seqa(1,1,k)';
    i=i+1;
enddo;

```

```

x=labelx | x;
x=x';
x=delif(x,(x[:,1] .== d));
x=x'; /* Try to see print x; */
x=trimr(x,1,0);

```

いうように、独立変数行列  $x$  に対する  $labelx$  というインデックス行を考えて、最初に 0、次に 3 変数の場合 1,2,3,1,2,3,... と続くようにインデックスをつけ  $x$  の上に 1 行加えます。それを転置した後、 $delif$  で  $x$  の 1 列目が  $d$  の中に入っている数である行を取り除きます。すなわち、もとの  $x$  のある列に相当するものが取り除かれます。それを再び転置して元の向きに戻した後に、 $trimr$  で上から 1 行インデックスに相当する部分をカットして、制限モデルの従属変数が完成します。これをもとに、制限モデルの RSS を求めます。最後に、これら 2 つの RSS を用いて、F 統計量を計算して、分子分母の値にもとづく P 値を計算してから、これを与えられた Significance Level の  $pp$  の 0.05 と比べて、小さければ  $H_0$  を棄却し、そうでなければ棄却しない条件分岐をさせています。これを  $grangermain$  として、前半の  $granger$  で、従属変数を  $c$  列目、独立変数のうち取り除くインデックス番号を  $d$  として、二重ループでそれぞれ変数の個数  $k$  までまわして、 $c \neq d$  の場合にだけ  $grangermain$  を呼び出して計算させ、結果をメッセージとともに画面表示させています。そうでない場合 ( $c=d$  のとき)には何もしないようにしています。これにより、すべてのケースについて計算できます。なお、ここでの Significance Level は 0.05 を採用しています。

#### 画面表示

```

1 <- 2
F(      2 ,      30 ) =      0.69396629 :P-value= 0.50743236

5      % significance level      NOT reject H0

1 <- 3
F(      2 ,      30 ) =      0.75815573 :P-value= 0.47729630

5      % significance level      NOT reject H0

2 <- 1
F(      2 ,      30 ) =      5.2483850 :P-value= 0.011104859

5      % significance level      Reject H0[ Granger's causality exists!]

```

```

2 <- 3
F(      2 ,      30 ) =      4.1809761 :P-value= 0.025021211

```

```

5      % significance level      Reject H0[ Granger's causality exists!]

```

```

3 <- 1
F(      2 ,      30 ) =      1.6968484 :P-value= 0.20037859

```

```

5      % significance level      NOT reject H0

```

```

3 <- 2
F(      2 ,      30 ) =      1.4998482 :P-value= 0.23942509

```

```

5      % significance level      NOT reject H0

```

上のプログラムは多変数の場合ですが、1対1の2変数を関係を見るのには、上のプログラムの呼び出しのところの3変数以上あるところを最初から例えばY~Kなどと2変数にすれば、通常の1対1の場合の Granger Test ができます。ここでは、

$$\begin{aligned}
 Y_t &= \beta_{11} + \beta_{12} Y_{t-1} + \beta_{13} L_{t-1} + \beta_{14} Y_{t-2} + \beta_{15} L_{t-2} + u_{1t} \\
 Y_t &= \beta_{21} + \beta_{22} Y_{t-1} + \beta_{24} Y_{t-2} + u_{2t}
 \end{aligned}$$

の非制限モデルと制限モデルを考えて、それぞれの残差平方和  $RSS_u$  と  $RSS_r$  を用いて、

$$F = \frac{(RSS_r - RSS_u) / 2}{RSS_u / (n - 5)} \sim F(2, n - 5)$$

でテストを行なうものです。ここでは  $\beta_{13}$  と  $\beta_{15}$  が0に制限されていますから自由度は分子が2、分母はもともとのパラメータの数は、ここではKのラグ項がなくて、5がきます。

プログラムの的には、データ行列のいくつかある列のうち2つだけを取り出して、その2変数だけにたいする Granger Test を前述の grangermain に代入して行なわせています。ですから、ここでは、異なる2変数の組合せをサンプルするアルゴリズムだけが必要になります。これは、cとdの二重ループでそれぞれ1からデータ行列の列数kまで動かしてやって、その二重ループの最深部に c=d となればメッセージとともに grangermain を呼び出すようにして、そうでなければ（すなわち c=d のケースには）何もしないような granger2 の proc を作成して、その第2引数のラグ数と第3引数の Significance Level については何もせずに、直接内部の grangermain の呼び出しに用います。

## プログラム

```
new; cls;
load data[40,3]=d:datafile13.txt;
Y=data[2:40,1]; L=data[2:40,2]; K=data[2:40,3];
data=Y~L~K;
call granger2(data,2,0.05);
```

```
proc(0)=granger2(data,lag,pp);
local k,c,d,data2;
k=cols(data);
c=1;
do while c<=k;
  d=1;
  do while d<=k;
    if c/=d;
      data2=data[.,c d];
      print "          =====";
      print/rz c "<-";;print/lz d;
      print "          =====";
      call grangermain(data2,lag,1,2,pp);
    endif;
    d=d+1;
  endo;
  c=c+1;
endo;
endp;
```

( この後に、proc grangermain を置く )

## 画面表示

```
=====
      1 <- 2
=====
F(      2 ,      32 ) =      0.15919461 :P-value= 0.85350160

5          % significance level      NOT reject H0
```

=====

1 <- 3

=====

F( 2 , 32 ) = 0.22528687 :P-value= 0.79954254

5 % significance level NOT reject H0

=====

2 <- 1

=====

F( 2 , 32 ) = 1.3185140 :P-value= 0.28167591

5 % significance level NOT reject H0

=====

2 <- 3

=====

F( 2 , 32 ) = 0.40555553 :P-value= 0.66998436

5 % significance level NOT reject H0

=====

3 <- 1

=====

F( 2 , 32 ) = 0.85683908 :P-value= 0.43401173

5 % significance level NOT reject H0

=====

3 <- 2

=====

F( 2 , 32 ) = 0.65795117 :P-value= 0.52477649

5 % significance level NOT reject H0

基本的に 1 階部分の grangermain には、 $N \times K$  のデータ行列の「c 列目への d 列目からの」Granger の意味での Causality を F 分布にもとづいて調べているのですが、その 2 階部分が granger2 とすると、grangermain を呼び出す際に data 行列そのものを  $N \times 2$  として 1 対 1 の変数系列を調べることにしています。なお、granger2 または granger(data,lag,pp) の pp のところに 5 % に相当する 0.05 ではなくて、0.01 や 0.1 を代入すると 1 % や 10 % に相当する結果が表示されます。第 2 要素の lag のところには、上のように固定ではなくて、データ行列から自動設定にするには、proc varlag をプログラム中のどこかに置いてから、

lag=varlag(data,opt);

などとしてやって、変数 lag の中味を受け渡せば、VAR の分析結果とともに Granger テストの結果が得られます。

### Sims Test

上のものと同じく、1 対 1 の変数同士の Causality を調べるのに、Sims のバージョンの Granger Test があります(ここでは先行ラグ数 2、定数項ありのケース)。

$$Y_t = \alpha_{11} + \alpha_{12} L_t + 2 + \alpha_{13} L_{t+1} + \alpha_{14} L_t + \alpha_{15} L_{t-1} + \alpha_{16} L_{t-2} + u_{1t}$$

$$Y_t = \alpha_{11} + \alpha_{14} L_t + \alpha_{15} L_{t-1} + \alpha_{16} L_{t-2} + u_{1t}$$

の非制限モデルと制限モデルを考えて、それぞれの残差平方和  $RSS_u$  と  $RSS_r$  を用いて、

$$F = \frac{(RSS_r - RSS_u) / 2}{RSS_u / (n - 6)} \sim F(2, n - 6)$$

をもとにテストするものです。ここでは、リードラグを除いた時の影響を

$$H_0: \alpha_{12} = \alpha_{13} = 0$$

$H_1$ :  $H_0$  is NOT true. (L から Y にリードラグの影響がある、Granger の因果性あり) という仮説でテストしています。なお、定数項なしにテストする方法もありますが、この例の場合、これとほとんど同じ結果になります。上と同じテストをほかの変数の組み合わせについても行ないます。

### プログラム

```
new; cls;
load data[40,3]=d:datafile13.txt;
Y=data[2:40,1]; L=data[2:40,2]; K=data[2:40,3];
data=Y~L~K;
call sims(data,2,0.05);
```

```
proc(0)=sims(data,lag,pp);
```



```

local k,c,d;
k=cols(data);
print "Sims Test";
c=1;
do while c<=k;
    d=1;
    do while d<=k;
        if c/=d;
            print "          =====";
            print/rz c "<-";;print/lz d;
            print "          =====";
            call simsmain(data,lag,c,d,pp);
        endif;
        d=d+1;
    endo;
    c=c+1;
endo;
endp;

```

```

proc(0)=simsmain(data,lag,c,d,pp);
    local n,k,i,xlag,yc,x,n1,k1,b,e,rssu,rssr,f,p;
    n=rows(data); k=cols(data);
    i=1; xlag=data[1:n-2*lag,d];
    do while i<=2*lag;
        xlag=xlag~data[1+i:n-2*lag+i,d];
        i=i+1;
    endo;
    yc=data[lag+1:n-lag,c];
    x=ones(n-2*lag,1)~xlag;
    n1=rows(x); k1=cols(x);
    b=inv(x'x)*x'yc;
    e=yc-x*b;
    rssu=e'e;
/* Restricted Model for Lead Lags */
    x=ones(n-2*lag,1)~xlag[.,1:lag+1];
    b=inv(x'x)*x'yc;

```

```

e=yc-x*b;
rssr=e'e;
/* Calculate F */
f=((rssr-rssu)/lag) / (rssu/(n1-k1));
print/rz "F(" lag ", " n1-k1 ") =";print f;;
p=cdfc(f,lag,n1-k1); print/lo ":P-value=" p;
print/lz pp*100 "% significance level      ";
if p<pp;
    print "Reject H0[ (Sims)Granger's causality exists!] ";
else;
    print "NOT reject H0";
endif;
print;
endp;

```

なお、Granger の意味での因果性を超えて、より意味のある変数間の因果関係を見るのには、後の章で扱うような相関係数の有意性からパターン分析をする Directed Graph Study のテクニックを用います。

### Block Exogeneity Test

理論があってモデルを検証するのではなくて、データにモデルを説明させるような VAR では、このブロック外生性でもってどの変数を除くべきか残すべきかをテストします。

非制限モデル

$$\begin{aligned}
 Y_t &= \pi_{11} + \pi_{12} Y_{t-1} + \pi_{13} L_{t-1} + \pi_{14} K_{t-1} + \pi_{15} Y_{t-2} + \pi_{16} L_{t-2} + \pi_{17} K_{t-2} + u_{1t} \\
 L_t &= \pi_{21} + \pi_{22} Y_{t-1} + \pi_{23} L_{t-1} + \pi_{24} K_{t-1} + \pi_{25} Y_{t-2} + \pi_{26} L_{t-2} + \pi_{27} K_{t-2} + u_{2t} \\
 K_t &= \pi_{31} + \pi_{32} Y_{t-1} + \pi_{33} L_{t-1} + \pi_{34} K_{t-1} + \pi_{35} Y_{t-2} + \pi_{36} L_{t-2} + \pi_{37} K_{t-2} + u_{3t}
 \end{aligned}$$

制限モデル

$$\begin{aligned}
 Y_t &= \pi_{11} + \pi_{12} Y_{t-1} + \pi_{13} L_{t-1} + \pi_{15} Y_{t-2} + \pi_{16} L_{t-2} + u_{1t} \\
 L_t &= \pi_{21} + \pi_{22} Y_{t-1} + \pi_{23} L_{t-1} + \pi_{25} Y_{t-2} + \pi_{26} L_{t-2} + u_{2t} \\
 K_t &= \pi_{31} + \pi_{32} Y_{t-1} + \pi_{33} L_{t-1} + \pi_{34} K_{t-1} + \pi_{35} Y_{t-2} + \pi_{36} L_{t-2} + \pi_{37} K_{t-2} + u_{3t}
 \end{aligned}$$

として、それぞれに対する variance covariance 行列 vcv を求めます。これらを と考えて、それをもとに、実質使われている行数 T に対して、Likelihood Ratio テストを

$$T(\ln |r| - \ln |u|) \sim \chi^2(df) \quad \text{ここで } df=(k-1)*lag$$

にもとづいて行ないます。ここでは k=3 と lag=2 ですから (3-1)\*2=4 の 4 の自由度に対してテストを行ないます。テストすべき仮説は、

H0:  $\beta_{14} = \beta_{17} = \beta_{24} = \beta_{27}$  (制約が正しい、取り除いた変数を取り除くべき)

H1: H0 is NOT true. (制約が正しくない、取り除いた変数は含まれるべき)

でカットオフ値を超えれば、H0 は棄却され、制約が正しくないことになります。すなわち、その場合、除かれた変数 K はモデルに含まれるべきと判断されます。

プログラム

```
new; cls;
load data[40,3]=d:datafile13.txt;
Y=data[2:40,1]; L=data[2:40,2]; K=data[2:40,3];
data=Y~L~K;
call blockf(data,2,0.05);
```

```
proc(0)=blockf(data,lag,pp);
local i;
print "Block Exogeneity Test";
print "      H0: Restrictions are well imposed.";
print "      H1: H0 is NOT true.";
i=1;
do while i<=cols(data);
    call blockfmain(data,lag,i,pp);
    i=i+1;
enddo;
endp;
```

```
proc blockfmain(data,lag,d,pp);
local n,k,n1,i,j,xlag,y,x,b,e,vcov,omegau,labelx,er,vcovr,omegar,lr,df,p;
n=rows(data); k=cols(data); n1=n-lag;
i=2; xlag=data[lag:n-1,.];
do while i<=lag;
    xlag=xlag~data[lag+1-i:n-i,.];
    i=i+1;
enddo;
y=data[lag+1:n,.];
x=ones(n-lag,1)~xlag;
b=inv(x'x)*x'y;
e=y-x*b;
vcov=e'e/n1;
```

```

    omegau=det(vcov);
/* Restricted Model */
    i=1; labelx=0;
    do while i<=lag;
        labelx=labelx~seqa(1,1,k)';
        i=i+1;
    endo;
    x=labelx | x;
    x=x';
    x=delif(x,(x[.,1] .== d));
    x=x';
    x=trimr(x,1,0);
    b=inv(x'x)*x'y;
    er=y-x*b;
    er[.,d]=e[.,d];
    vcovr=er' er/n1;
    omegar=det(vcovr);
/* Calculate LR */
    lr=n1*(ln(omegar)-ln(omegau));
    print/lz "Variable" d;
    df=(k-1)*lag;
    print "LR=" lr;;print/rz "~X2(" df ")";;
    p=cdfchic(lr,df); print/ld ":P-value=" p;
    print/lz pp*100 "% significance level    ";;
    if p<pp;
        print "Reject H0";
    else;
        print "NOT reject H0";
    endif;
    retp(lr);
endp;

```

上のプログラムも Granger Test の時と基本的に同じですが、制限モデルの方は、最初から 2 変数のモデルで推定して、上の場合第 3 式は非制限モデルのままですから、 $er[.,d]=e[.,d]$  として、非制限モデルの残差行列の第 d 列目を制限モデルの残差行列の第 d 列目に戻してやる作業をします。あとは、vcv まで求めて、それをもとに、Likelihood Ratio を計算して

やり 2(df)の分布にしたがうテストをしてやります。この proc を上の blockf の中で第 1 変数から最終 data の列数の番目の変数までのすべての場合について呼び出して計算させています。

#### 画面表示

##### Block Exogeneity Test

H0: Restrictions are well imposed.

H1: H0 is NOT true.

##### Variable 1

LR= 36.999583 ~X2( 4 ):P-value= 0.00000018  
5 % significance level Reject H0

##### Variable 2

LR= 13.298092 ~X2( 4 ):P-value= 0.00990748  
5 % significance level Reject H0

##### Variable 3

LR= 27.465853 ~X2( 4 ):P-value= 0.00001600  
5 % significance level Reject H0

当然のことながら、結果はすべての変数のケースについて H0 は棄却されて、どの変数も取り除かずに残しておくべきだと判断できます。