3.21　VAR モデル(3)　　ver.0.1 工事中

ここでは、VAR で計算された誤差項の分散共分散行列を用いて、まず、相関係数とその Z そしてその P 値を求めます。同様に、偏相関係数についても計算します。そして、最後に、それらのパターンを見て行なう Directed Graph の因果関係を示します。

Unconditional and Partial Correlation

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;
    ddata=data[2:rows(data),.]-data[1:rows(data)-1,.];
    call varcorr(ddata,1);


proc(6)=varcorr(data,opt);
    local maxlag,crit,aic,sc,hq,fpe,j,i,n,k,t,np,xlag,y,x,b,e,s2mat,detvcov,ll,lag;
    local vcv,se,corr,c,parcorr,z0,z1,nused,order,p0,p1;
/* Lag Order to min information criteria*/
    /* opt= 1 for min AIC,   2 for min SC,   3 for min HQ. */
    if opt==1;
        print "Min AIC criterion:";
    elseif opt==2;
        print "Min Schwarz information criterion:";
    elseif opt==3;
        print "Min Hannan-Quinn information criterion:";
    else;
        errorlog "ERROR:Opt # must be 1:AIC or 2:SC or 3:HQ.";
        retp(-1);
    endif;
    maxlag=1;
    do while maxlag<=24;
        if rows(data)-maxlag-cols(data)*maxlag-1<cols(data);
        @   d.f. is set to be more than # of variables. @
            break;
        endif;
        maxlag=maxlag+1;
```

```
    endo;
    maxlag=maxlag-1;
    crit=zeros(maxlag,3);
    j=1;
    do while j<=maxlag;
        n=rows(data); k=cols(data); t=n-j; np=j*(1+k*j);
        i=2; xlag=data[j:n-1,.];
        do while i<=j;
            xlag=xlag~data[j+1-i:n-i,.];
            i=i+1;
        endo;
        y=data[j+1:n,.];
        x=ones(n-j,1)~xlag;
        b=inv(x'x)*x'y;
        e=y-x*b;
        s2mat=e'e/(n-j);
        detvcov=abs(det(s2mat));
        if detvcov<1e-16;   /* Avoid 0 determinant. */
            detvcov=1e-16;
        endif;
        ll=-k*t/2*(1+ln(2*pi))-t/2*ln(detvcov);
        aic=-2*ll/t+2*np/t;
        sc=-2*ll/t+np*ln(t)/t;
        hq=-2*ll/t+2*np*ln(ln(t))/t;
        crit[j,.]=aic~sc~hq;
        j=j+1;
    endo;
    print "       LAG                AIC                SC                HQ";;
    print seqa(1,1,maxlag)~crit;
    lag=minindc(crit[.,opt]);
    print;
    print/rz "Choice of LAG #" lag;
/* VAR estimation to get VCV*/
    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;
    b=inv(x'x)*x'y;
    print "b:" b;
    e=y-x*b;
    vcv=e'e/(n-lag);
    se=sqrt(diag(inv(x'x))*diag(vcv)');
    t=b./se;
/* Correlation */
    @ Or, corr=corrvc(vcv); @
    corr=zeros(k,k);
    i=1;
    do while i<=k;
        j=1;
        do while j<=k;
            corr[i,j]=vcv[i,j]/(sqrt(vcv[i,i])*sqrt(vcv[j,j]));
            j=j+1;
        endo;
        i=i+1;
    endo;
/* Partial Correlation */
    c=inv(corr);
    parcorr=zeros(k,k);
    i=1;
    do while i<=k;
        j=1;
        do while j<=k;
            parcorr[i,j]=c[i,j]/(sqrt(c[i,i])*sqrt(c[j,j]));
            j=j+1;
        endo;
        i=i+1;
    endo;
    parcorr=-parcorr;
    i=1;
```

```
    do while i<=k;
        parcorr[i,i]=-parcorr[i,i];
        i=i+1;
    endo;
/* Z and its P-value */
    nused=n-lag;
    z0=1./zeros(k,k); z1=1./zeros(k,k);
    i=1;
    do while i<=k;
        j=1;
        do while j<=k;
            if i/=j;
                order=0;
                z0[i,j]=0.5*sqrt(nused-order-3)*ln((1+corr[i,j])/(1-corr[i,j]));
                order=k-2;
                z1[i,j]=0.5*sqrt(nused-order-3)*ln((1+parcorr[i,j])/(1-parcorr[i,j]));
            endif;
            j=j+1;
        endo;
        i=i+1;
    endo;
    p0=2*cdfnc(abs(z0)); p1=2*cdfnc(abs(z1));
/* Display Results */
    print; print "Variance Covariance Matrix from VAR Residuals";
    print "vcv:" vcv;
    print/rz "N used:" nused;
    print; print "Unconditional Correlation";
    print "corr:" corr;
    print/rd "p-value:" p0;
    print; print "Conditional Correlation";
    print "partial corr:" parcorr;
    print/rd "p-value:" p1;
    retp(vcv,nused,corr,parcorr,p0,p1);
endp;
```

画面表示

Min AIC criterion:

LAG	AIC	SC	HQ
1.0000000	18.282584	18.456737	18.343981
2.0000000	18.087125	18.702938	18.302060
3.0000000	18.834016	20.167171	19.294221
4.0000000	19.991213	22.325647	20.787322
5.0000000	21.117223	24.745121	22.337900
6.0000000	22.961793	28.183477	24.692633
7.0000000	25.096534	32.220213	27.418676
8.0000000	25.675824	35.017140	28.664191

Choice of LAG # 2

b:

6.9607947	2.9751673	2.0938801
0.059151587	-0.016030269	-0.20646028
-0.50566020	-0.056169881	0.12189057
1.6726696	0.59394204	0.95038666
0.48186923	0.50952775	0.20496263
-0.19541769	-0.38745010	-0.31405757
-1.1082540	-1.3771054	-0.15801375

Variance Covariance Matrix from VAR Residuals

vcv:

144.21918	60.869195	36.726473
60.869195	35.533643	16.916487
36.726473	16.916487	14.209708

N used: 36

Unconditional Correlation

corr:

1.0000000	0.85028827	0.81128817
0.85028827	1.0000000	0.75283141
0.81128817	0.75283141	1.0000000

p-value:

0.00000000	0.00000000	0.00000000

```
       0.00000000        0.00000000        0.00000002
       0.00000000        0.00000002        0.00000000

Conditional Correlation
partial corr:
        1.0000000        0.62243055        0.49408225
       0.62243055         1.0000000        0.20474697
       0.49408225        0.20474697         1.0000000
p-value:
       0.00000000        0.00003729        0.00219214
       0.00003729        0.00000000        0.24006345
       0.00219214        0.24006345        0.00000000
```

Directed Acyclic Graph

上の結果を 10%の有意水準で判断するとすると、まず相関係数が０であることを棄却できるものの間のエッジを取り除きます。この場合、ありません。次に、偏相関係数が０であることを棄却できるもののエッジを取り除きます。この場合、ΔＬとΔＫの間のエッジを取り除きます。その結果、ΔＬ－ΔＹ－ΔＫという３者のひと続きの関係になります。ここで、corr(ΔＬ,ΔＫ)は０ですから、まんなかのΔＹはΔＬ とΔＫのセップセットにあるのでエッジに対して矢印が書くことができません。ΔＬ－ΔＹ－ΔＫという関係はありますが、矢印の方向はその他の情報によって決まります。

プログラム

画面表示

４変数以上の場合

ここで、０と１からなる nCr の組み合わせのインデックス行の集まりの行列を生成する procedure をあらかじめ作成しておきます。これにより、複数変数の組合せのピックアップがより効率的になります。

プログラム

```
new; cls;
print/rz combi01(5,2);

/*
** Binary sequence matrix of nCr combination
```

```
** (C) Copyright 2002 Yosuke Amijima. All Rights Reserved.
**
** PROC COMBI01
**
** FORMAT
**          comb = combi01(n,r)
** INPUT
**          n - total number considered. We here think of nCr.
**          r - number of times taken from total number.
** OUTPUT
**          comb - nCr x n matrix(nCr row vectors of binary sequence).
** Remarks:
**          This algorithm is pretty simple and short, but it really takes
**          lots of memory to run. GAUSS Light would work only when n<=9.
*/
proc combi01(n,r);
    local comb,x,i,temp,index;
    comb=zeros(2^n,n);
    x=seqa(1,1,2^n);
    i=1;
    do while i<=n;
        temp=x;
        x=floor(x/2);
        comb[.,i]=temp-2*x;
        i=i+1;
    endo;
    index=zeros(2^n,1);
    i=1;
    do while i<=2^n;
        if sumc(comb[i,.]')==r;
            index[i]=1;
        endif;
        i=i+1;
    endo;
    comb=selif(comb,index);
    retp(comb);
```

endp;
上のアルゴリズムは極めてシンプルです。ｎ個の０と１からなる数列を考えて、そのすべての場合は当然２のｎ乗通りあります。これについて、ずらしていくことによって、それらすべての Binary の数列をまず出します。これを行方向（横方向）に見て、１の合計がｒになるものだけをインデックス１に設定して、それを selif で論理によりピックアップしています。これにより、横方向に１がｒ個だけあるすべての Binary 数列のすべての組合せが簡単に得られます。（ただし、２のｎ乗すべてを先に展開してから、絞り込む形をとっているのでかなりのメモリーを消費します。上の場合、n=9 程度までしか Light 版では動きません。）

画面表示

1	1	0	0	0
1	0	1	0	0
0	1	1	0	0
1	0	0	1	0
0	1	0	1	0
0	0	1	1	0
1	0	0	0	1
0	1	0	0	1
0	0	1	0	1
0	0	0	1	1

当然のことながら、行数は ${}_5C_2=10$ となります。１番目と２番目の２つを選択する時、１番目と２番目は１で、その他は０になります。それが、第１行目になっています。以下、１番目と３番目を選択する場合が２行目、最後の行は、４番目と５番目を選択する場合にそれぞれ対応しています。
プログラム

画面表示

アルゴリズムは、４変数以上の場合特に、かなり大きな数の変数になったときが問題で、上の相関、変相関行列の Z とそれに対する P 値求めるのですが、上の場合、０次と最終次だけが求まっていることになります。３変数の場合はこれだけで完了ですが、変数が増すにしたがって、エッジが取れていない４変数、それでも取れなければ５変数というふうにすべての場合を計算する必要があります。単純なのですが、かなりの繰り返しをどう減らすのか、現在 efficient なアルゴリズムを考案中です。