

3.10 プログラミング 構造変化とテスト

ver.0.2

最近最もよく使われるようになった時系列の構造変化の諸テストを扱います。Time Seriesのデータに対するOLSで、構造変化の時点が既知でない場合には、F分布を用いた通常のChowテストでは観測できません。変化時点が既知でない場合には、 $t+1, t+2, \dots$ と繰り返し推計によるChowテストも考えられますが、その統計値はF分布ではなくなってしまうので厳密には有効ではありません。そこでその代わりに考えるのが、時点 t 期に、 $t-1$ 期までのデータを用いた予測値を用いて、 t 期の実測値と比較したrecursive residualsを求めてそれをstandardizedしたものを統計量にしてテストしようとする考え方です。

$$\text{Recursive residual: } e_t = y_t - x_t' \beta_{t-1} \quad t=k+1, k+2, \dots, n$$

ただし、 β_{t-1} は $t-1$ 期までの情報によって計算された係数

ここで、最終 n 期まで構造変化がないと仮定するならば、

$$E(e_t) = 0 \quad \text{and} \quad \text{Var}(e_t) = \sigma^2 s(e_t)^2$$

であるので、standardizedするために、この e_t を $s(e_t)$ で割ってやれば

$$w_t = e_t / s(e_t) \sim N(0, \sigma^2) \quad t=k+1, k+2, \dots, n$$

にしたがうようになる。この w_t を standardized recursive residual といって、これから行なうほとんどのテストのベースになる統計量です。これを、従属変数ベクトル y と独立変数行列 x が与えられているときに計算するprocedureは次のようになります。

プログラム

```
proc wstat(y,x);
  local n,k,w,ylag,xlag,bhatlag,e,t,v;
  n=rows(x); k=cols(x);
  w=zeros(n-k,1);
  t=k+1;
  do while t<=n;
    xlag=x[1:t-1,.]; ylag=y[1:t-1,.];
    bhatlag=inv(xlag'xlag)*xlag'*ylag;
    e=y[t,]-x[t,]*bhatlag;
    v=1+x[t,]*inv(xlag'xlag)*x[t,]';
```

```

w[t-k,]=e/sqrt(v);
t=t+1;
endo;
retp(w);
endp;

```

このporcedureでは、まず、それぞれのデータシリーズの個数 n と、 x 行列の列の個数 k を求めます。そして、いま作ろうとしているstandardized recursive residualの w_t は、列ベクトルで、 $k+1$ から n までの $n-k$ 個ですから初期値として、 $n-k \times 1$ の零ベクトルを作成してこれを w とします。それぞれの t 期に対してその期ごとに、1 期から $t-1$ 期までのデータをもとに係数 $\hat{\beta}$ を求めます。これを x_t とともに使って、 y_t の予測値を求めます。実測値 y_t とこの予測値の差が、 e のrecursive residualになります。これをstandardizedするには、この場合の $s(e_t)^2$ は、行列計算で、

$$s(e_t)^2 = 1 + x_t(x_{-1}'x_{-1})^{-1}x_t' \quad \text{ただし、} x_{-1} \text{ は } t-1 \text{ 期までのすべてのデータ}$$

ですので、このルートをとった値で割って、 e をstandardizedします。Doループの中は、この t 期に対する作業を、初期値の $k+1$ 期から 1 ごと期を増やして、最終的に n 期まで行なうものです。そして、リターンは w の列ベクトルを返します。このとき、 $k+1$ 期から始まることに注意してください。説明変数 1、定数項 1 の場合には、 $k=2$ ですので、3 期から始まることになります。以下の例では、 $k=5$ について求めてみます。

CUSUM Test

これに対して、Dドライブにdatafile11.txtという50行5列のデータがあるものとします。1列目を y 、2列目から5列目を x として、定数項となる1の列ベクトルをつけます。すなわち、 x は 50×5 のディメンションになります。そして、このデータをOLSをもとにしてstandardized recursive residualの w_t を上述のproc wstatで計算したうえで、CUSUMテストをしてみます。この場合のテスト統計量は、

$$W_j = \frac{\sum_{t=k+1}^j w_t}{\hat{\sigma}} \quad j=k+1, k+2, \dots, n \quad \text{ただし、} \hat{\sigma} \text{ は } w \text{ の標準偏差}$$

j 期の上限と下限は、

$$w^* = \pm \frac{\theta \sqrt{n-k} + 2\theta(j-k)}{\sqrt{n-k}} \quad \text{となる。}$$

ここで、 θ は0.948(5%)、0.850(10%)とわかっている。ここでは、5%に対する値を使うことにする。 W_j のなかに、この上限と下限を超えるものがあれば、

H0: 構造変化がない

H1: H0 is NOT true.

という帰無仮説が棄却されて、構造変化があるとできる。すべての系列が上限と下限のあいだに位置していれば、構造変化があるとは言えないことになる。通常はグラフの分析と組になることが多い。上限と下限は0を中心にして期が進むにつれて放射線状に広がっていく形状をとる。このなかから系列が出てしまえば、構造変化はないとする帰無仮説は棄却されることになる。

なお、実際のプログラムでは、下のように、上で作成したwstatを置いて、その下に新たに、cusumtestというprocedureを作成して、そのprocedureにwを受け渡して、冒頭で2つのprocedureを呼び出すことによって計算をさせています。cusumtestのprocedureは4つのリターンを返すものにしてあって、それを冒頭で水平方向に4つをマージして表示しています。これに対応するCUSUMのグラフは、cusumtestのprocedureの内部でpgraphのライブラリを呼び出して描画を行なっています。

プログラム

```
new; cls;
load data[50,5]=d:datafile11.txt;
y=data[,1]; x=ones(rows(data),1)~data[,2:5];
w=wstat(y,x);
n=rows(x); k=cols(x); theta=0.948; @ At 5% @
{j,wmin,wj,wmax}=cusumtest(w,n,k,theta);
print j~wmin~wj~wmax;
```

```
proc wstat(y,x);
  local n,k,w,ylag,xlag,bhatlag,e,t,v;
  n=rows(x); k=cols(x);
  w=zeros(n-k,1);
  t=k+1;
  do while t<=n;
    xlag=x[1:t-1,.]; ylag=y[1:t-1,.];
    bhatlag=inv(xlag'xlag)*xlag'*ylag;
    e=y[t,.]-x[t,]*bhatlag;
    v=1+x[t,]*inv(xlag'xlag)*x[t,]';
    w[t-k,.]=e/sqrt(v);
    t=t+1;
  endo;
```

```

        retp(w);
    endp;

proc (4) = cusumtest(w,n,k,theta);
    local j,wj,wmin,wmax,theta,wstar;
    j=seqa(k+1,1,n-k);
    wj=cumsumc(w)./stdc(w);
    wmax=theta*sqrt(n-k)+2*theta*(j-k)/sqrt(n-k);
    wmin=-theta*sqrt(n-k)-2*theta*(j-k)/sqrt(n-k);
    wstar=maxc(abs(wj*sqrt(n-k)/((n-k)+2*(j-k))));
    if wstar>theta;
        print "CUSUM test(RESULT:Reject H0)";
    else;
        print "CUSUM test(RESULT:Not reject H0)";
    endif;
    library pgraph;
    graphset;
    _pltype={1,1,6};
    title ("CUMSUM TEST"); xlabel ("time j"); ylabel ("CUSUM");
    xy(j,wmin~wmax~wj);
    retp(j,wmin,wj,wmax);
endp;

```

上のように、cusumtestのprocedureはwstatから受け渡されるwをもとにn,k,thetaを既知のものとして計算が進められます。nはxの行数、kはxの列数（すなわち、定数項を含む説明変数の数）です。ここで、jはwの系列と一致するだけの期のラベルで、k+1からnまでの1刻みの数の列です。これはn-k個あり、グラフの横軸の期を表すラベルに使います。wjはj期のCUSUM統計量、wmaxとwminはその上限と下限です。統計量wjのすべての値が上限と下限の中におさまっていれば、Not rejectで、そうでなければRejectのメッセージを出します。次に、pgraphのライブラリを呼び出して、そのグローバル変数をリセットし、線の種類の指定を1番目と2番目は破線の1で、3番目は実線で描くようにグローバル変数_pltypeで3×1のベクトルで指定します。タイトル、xのラベル、yのラベルをそれぞれ、引用符で包んで文字列として指定します。そして最後に、x y グラフで、x 軸側に作成した期jのラベルを、y 軸側には、下限、上限、そしてwjの系列を水平方向にマージしたものを指定します。そして、procedureとしてのリターンには、j、wmin、wj、それにwmaxをあてます。冒頭で実際の数値の表示に利用することになります。なお、nとkもwstatでリターンとしてかえして受け渡すという方法もありますが、ここでは、wstatの

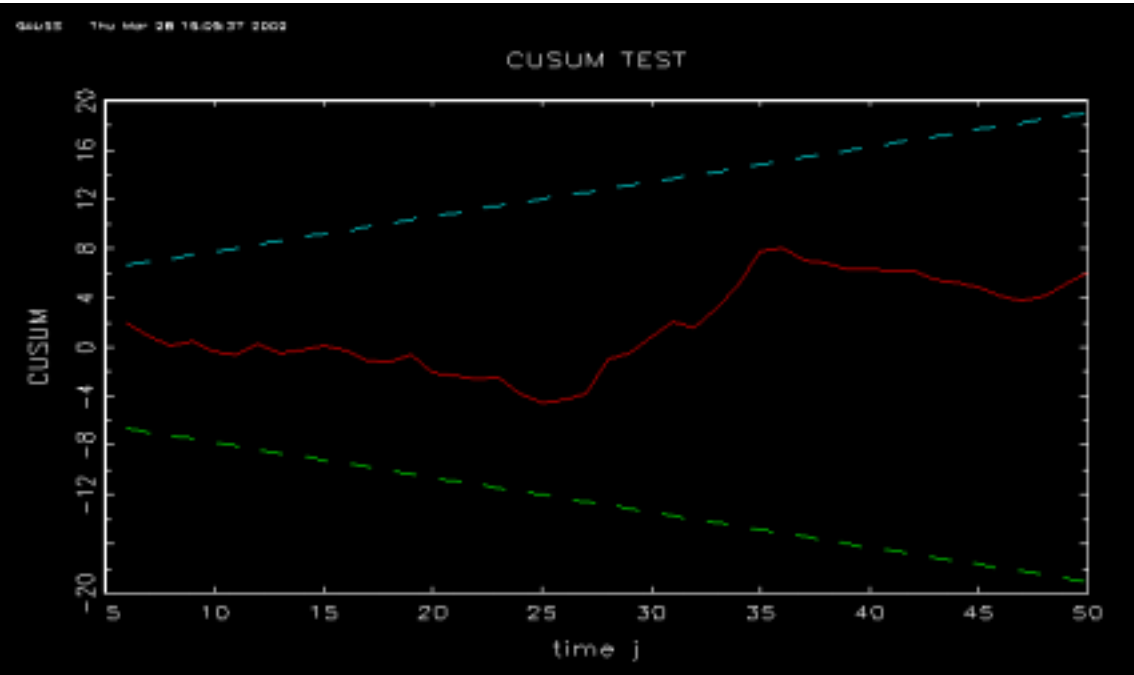
procedureを分かりやすい形にするためにわざとそうはしていません。若干の重複がプログラムに存在します。j 時点までの累積和を作るにはcumsumcという列のその行までの累積和を計算する組込み関数を使います。結果は以下ようになります。(点線部は間を省略してあります。)

画面表示

CUMSUM test(RESET:Not reject H0)

6.0000000	-6.6420163	1.9847910	6.6420163
7.0000000	-6.9246553	0.81013320	6.9246553
8.0000000	-7.2072943	0.087021022	7.2072943
.....			
48.000000	-18.512854	4.1608051	18.512854
49.000000	-18.795493	5.0430990	18.795493
50.000000	-19.078132	6.0208349	19.078132

グラフ表示



グラフから25期前後から若干キンクしていますが、すべての軌跡が上限下限の線の内側におさまっているので、構造変化の基準を満たすまでの変化にはなっていないことがわかります。

CUSUMSQ Test

同様に、同じデータ、同じstandardized recursive residualの w_t に対して、CUSUM of Squares Testをやってみます。テスト統計量は、今度は、

$$Q_j = \frac{\sum_{t=k+1}^j w_t^2}{\sum_{t=k+1}^n w_t^2} \quad j=k+1, k+2, \dots, n$$

を使います。すなわち、standardized recursive residualの w_t の全系列の二乗和に対するその時点 j までの二乗和の割合を統計量にします。最終的に n 期には、この統計量は、分母分子が等しくなって 1 になります。これに対する上限下限には、

$$q^* = \pm c + \frac{j-k}{n-k}$$

が使われます。グラフでは、原点(0,0)と(1,1)を結ぶ線に平行な上限と下限の領域になります。ただし、5%の有意水準では、 $c=0.25379$ ということが知られています。以下のプログラムは、CUSUMSQのプログラムの部分の統計量と上限下限の計算の部分を変更して簡単にできます。上限と下限のあいだの領域から1つでも超える Q_j の系列が含まれているのなら、

H0: 構造変化がない

H1: H0 is NOT true.

において、帰無仮説は棄却されて、構造変化があるということになります。反対に、すべての Q_j の系列が上限と下限のあいだにおさまっていれば、帰無仮説は棄却されなくて、構造変化はないということになります。

プログラム

```
new; cls;
load data[50,5]=d:datafile11.txt;
y=data[:,1]; x=ones(rows(data),1)~data[:,2:5];
w=wstat(y,x);
n=rows(x); k=cols(x); c=0.25379; @ At 5% @
{j,qmin,qj,qmax}=cumsumsqtest(w,n,k,c);
print j~qmin~qj~qmax;
```

```
proc wstat(y,x);
local n,k,w,ylag,xlag,bhatlag,e,t,v;
n=rows(x); k=cols(x);
w=zeros(n-k,1);
t=k+1;
```

```

do while t<=n;
    xlag=x[1:t-1,.]; ylag=y[1:t-1,.];
    bhatlag=inv(xlag'xlag)*xlag'*ylag;
    e=y[t,]-x[t,]*bhatlag;
    v=1+x[t,]*inv(xlag'xlag)*x[t,]';
    w[t-k,]=e/sqrt(v);
    t=t+1;
endo;
retp(w);
endp;

proc (4) = cusumsqtest(w,n,k,c);
    local qj,qmin,qmax,qstar,j,qstar;
    j=seqa(k+1,1,n-k);
    qj=cumsumc(w^2)/sumc(w^2);
    qmax=c+(j-k)/(n-k);
    qmin=-c+(j-k)/(n-k);
    qstar=maxc(abs(qj-(j-k)/(n-k)));
    if qstar>c;
        print "CUSUMSQ test(REJECT:H0)";
    else;
        print "CUSUMSQ test(REJECT:Not reject H0)";
    endif;
    library pgraph;
    graphset;
    _pltype={1,1,6};
    title ("CUSUMSQ TEST"); xlabel ("time j"); ylabel ("CUSUMSQ");
    xy(j,qmin~qmax~qj);
    retp(j,qmin,qj,qmax);
endp;

```

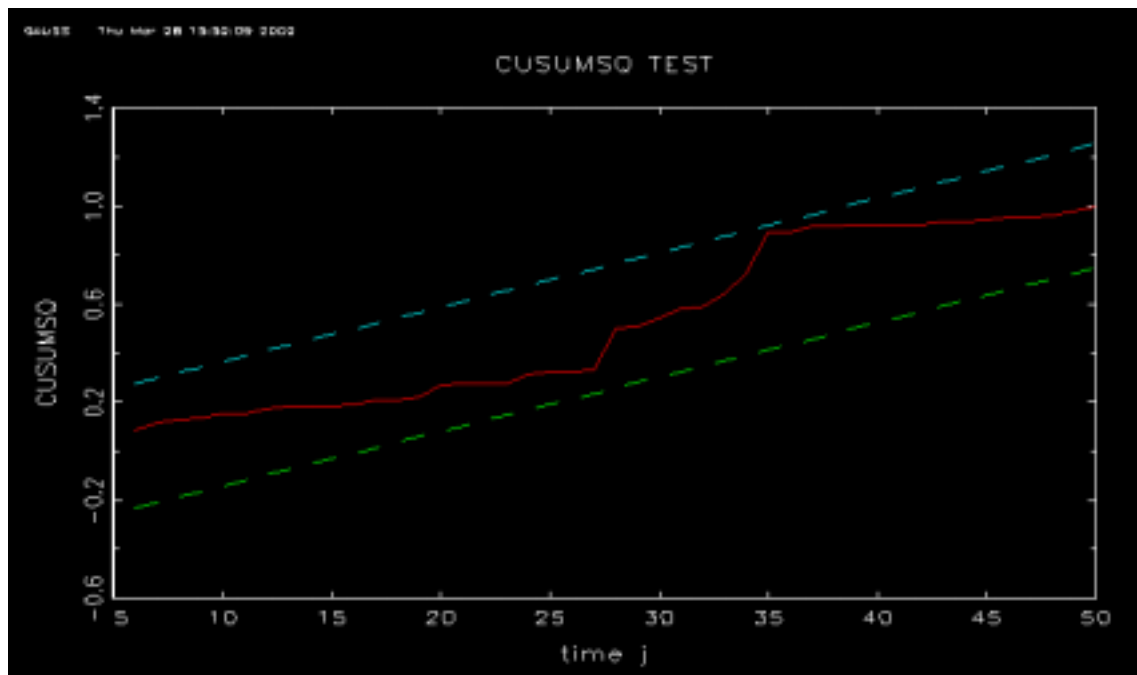
画面表示

CUSUMSQ test(REJECT:Not reject H0)

6.0000000	-0.23156778	0.087922009	0.27601222
7.0000000	-0.20934556	0.11871776	0.29823444
8.0000000	-0.18712333	0.13038799	0.32045667

48.000000	0.70176556	0.96129030	1.2093456
49.000000	0.72398778	0.97866409	1.2315678
50.000000	0.74621000	1.00000000	1.2537900

グラフ表示



グラフでは、 $n=50$ のとき統計量は1になっていることに注意してください。上のプログラムでも、 $j \sim q_{\min} \sim q_j \sim q_{\max}$ というふうにcusumstestの4つのリターンを水平方向にマージして画面表示させています。なお、この場合も、cusumcという列の累積和を求める組込み関数を使っています。テストの名前と混同することのないようにしてください。すなわちCUSUM TestもCUSUMSQ Testもこの累積和に関係するものであるということがわかったでしょうか。そういうわけで、cusumcという組込み関数は実はGAUSSの計算では重要なものとなっています。上のデータのケースでは、CUSUMでもCUSUMSQでもかろうじて5%の有意水準では、構造変化がないという帰無仮説は棄却されません。すなわち、構造変化はないと言えます。(なお、グラフでは上限と下限が1.2と-0.2まで伸びてしまっていますが、厳密には1と0を超える部分では境界線はキンクして1と0に水平になります。)

CUSUM Test with OLS residuals

このケースは、上の2つのケースと異なり、standardized recursive residualは計算しません。代わりに、通常のOLSのresidualを用いて、テスト統計量が

$$w_j = \frac{\sum_{t=1}^j e_t}{\hat{s}\sqrt{n}} \quad j=1,2, \dots, n$$

として、これに対しての5%のとき上限下限を決める値が、 $pk=1.358$ が知られていて、これにプラスマイナスをつけた範囲が上限下限になります。下のprocedureでは、jを1からnまでの列ベクトルの期を表すラベルとします。通常どおりOLSのresidualのeの列をもとめて、計算をすすめています。統計量wjの絶対値がpkの範囲にあるならば、Not rejectをメッセージとして出し、そうでなければRejectとします。さらに、pgraphのライブラリを呼び出して、グローバル変数を初期化し、CUSUMなどと全く同じようなグラフをxyグラフで描かせています。なお、proc wstatは計算上必要ないので書き加えていません。代わりに、OLSのresidualのeを使っているのがこのテストなのです。そして、冒頭でこの関数cusumtestolsを呼び出して、そのリターンのjとwjを水平方向にマージしたものをメッセージ $pk=1.358$ とともに表示させています。通常はグラフ表示はしないことが多いですが、あえてCUSUMおよびCUSUMSQとの比較で表示させると、5%に対する領域 $pk=1.358$ のプラスマイナスに対して、 $j = 1, 2, \dots, n$ の系列が描かれることになります。

プログラム

```
new; cls;
load data[50,5]=d:datafile11.txt;
y=data[.,1]; x=ones(rows(data),1)~data[.,2:5];
pk=1.358; @ At 5% @
{j,wj}=cusumtestols(y,x,pk);
print "pk=1.358" j~wj;

proc(2) = cusumtestols(y,x,pk);
    local j,n,k,wmax,wmin,b,e,pk,wj,shat;
    n=rows(x); k=cols(x);
    j=seqa(1,1,n); wmax=ones(n,1)*pk; wmin=-wmax;
    b=inv(x'x)*x'y;
    e=y-x*b;
    shat=sqrt(e'e/(n-k));
    wj=cumsumc(e)/(shat*sqrt(n));
    if abs(wj)<=pk;
        print "CUSUM test with OLS residuals (RESULT:Not reject H0)";
    else;
        print "CUSUM test with OLS residuals (RESULT:Reject H0)";
    endif;
```

```

library pgraph;
graphset;
_pltype={1,1,6};
title ("CUSUM w/ OLS residuals TEST");
xlabel ("time j"); ylabel ("CUSUM w/ OLS residuals");
xy(j,wmax~wmin~wj);
retp(j,wj);
endp;

```

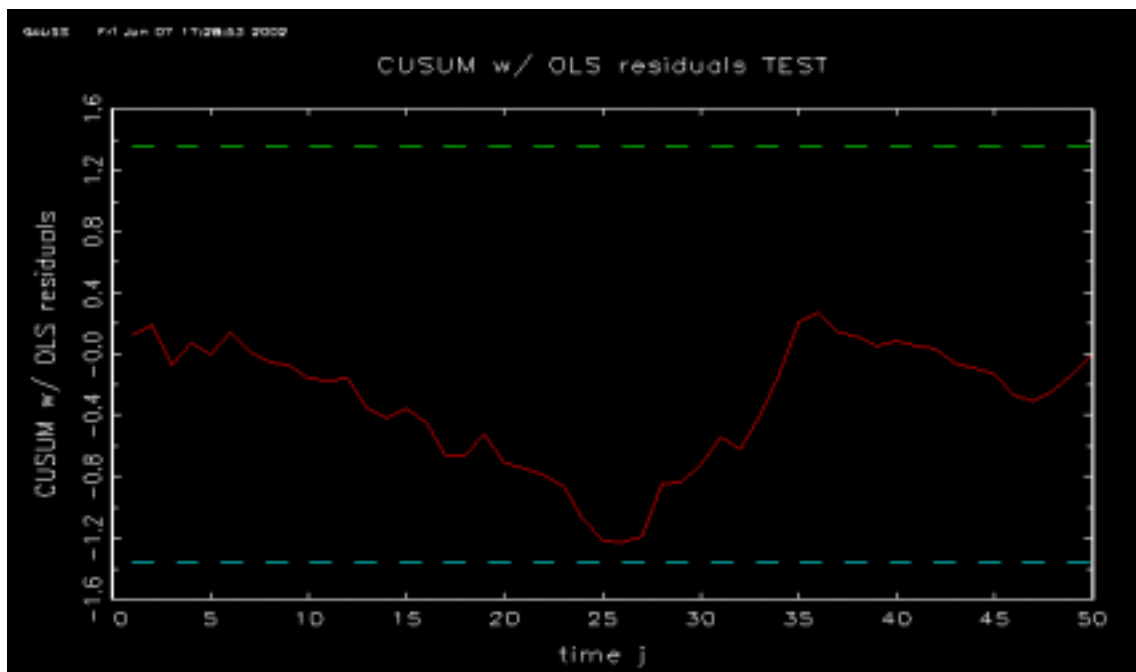
画面表示

CUSUM test with OLS residuals (RESULT:Not reject H0)

pk=1.358

1.0000000	0.12773502
2.0000000	0.18870783
3.0000000	-0.074337064
.....	
48.000000	-0.23855452
49.000000	-0.12884085
50.0	-1.1682040e-013

グラフ表示



Harvey-Collier t-test

このケースは、再びstandardized recursive residualの w_t に戻ってこれを用いて、trueな関係がリニアでないなら、 w_t のmeanは0から有意に異なっているということを利用してテストをします。この場合のstandardized recursive residualの w_t にもとづくテスト統計量は、

$$t = \frac{\frac{\text{mean}(w)}{\text{std}(w)}}{\sqrt{n-k}} \quad \text{ただし、mean}(w)\text{は}w\text{の平均、std}(w)\text{は}w\text{の標準偏差}$$

となるものを、t分布の complement のp値と自由度が与えられた時の値を求める cdfciを用いてカットオフポイントを計算します。5%の場合ならば、 $t > \text{cdfci}(0.025, n-k-1)$ または $t < \text{cdfci}(0.975, n-k-1)$ が棄却域になります。この場合のtはスケーラーです。すなわち、この2つのカットオフポイントよりも外側にあれば、構造変化がないという帰無仮説は棄却されることになり、反対に、そうでなければ帰無仮定は棄却されないことになります。

ポイント cdfci(p値,自由度) t分布にもとづく complement でその地点のt値を返す

プログラムの、このテストはグラフを書くことができません。wの平均値を使っているからです。スケーラーの統計量とt値にもとづく有意水準のカットオフポイントを比べて構造変化があるのかないのかだけを判別します。いつから構造変化があったかはこのテストでは判別はできません。下のプログラムの中では、冒頭で、proc hcttest を呼び出すのにリターンのtは必要ないので、call文でとめてリターンをなくしています。

プログラム

```
new; cls;
load data[50,5]=d:datafile11.txt;
y=data[.,1]; x=ones(rows(data),1)~data[.,2:5];
w=wstat(y,x);
n=rows(x); k=cols(x); p=0.05; @ AT 5% @
call hcttest(w,n,k,p);

proc wstat(y,x);
  local n,k,w,ylag,xlag,bhatlag,e,t,v;
  n=rows(x); k=cols(x);
  w=zeros(n-k,1);
  t=k+1;
```

```

do while t<=n;
    xlag=x[1:t-1,.]; ylag=y[1:t-1,.];
    bhatlag=inv(xlag'xlag)*xlag'*ylag;
    e=y[t,]-x[t,]*bhatlag;
    v=1+x[t,]*inv(xlag'xlag)*x[t,]';
    w[t-k,]=e/sqrt(v);
    t=t+1;
endo;
retp(w);
endp;

proc hcttest(w,n,k,p);
    local t;
    t=meanc(w)*sqrt(n-k)/stdc(w);
    if t>cdftci(p/2,n-k-1) or t<cdftci(1-p/2,n-k-1);
        print "Harvey and Collier t-test  t = " t " (RESULT:Reject H0)";
    else;
        print "Harvey and Collier t-test  t = " t " (RESULT:Not reject H0)";
    endif;
    print "Its cutoff points are      +-"; print cdftci(p/2,n-k-1);
    retp(t);
endp;

```

画面表示

```

Harvey and Collier t-test  t =      0.89753308  (RESULT:Not reject H0)
Its cutoff points are      +-      2.0153676

```

この結果からわかることは、構造変化がないとする帰無仮説が棄却されないので、wの平均値を用いた Harvey and Collier t-test でも、全体として、構造変化は起きていないと言えます。

Fluctuation Test

1つのFluctuation Testのバージョンとして、統計量を

$$ft = \frac{j}{\hat{sn}} \max \left| chol(x'x)(\hat{\beta}_j - \hat{\beta}) \right| \quad j = k+1, k+2, \dots, n$$

を計算して、5 %の時のカットオフポイント1.624と比べるというものがある。これを超えたら、構造変化はないという帰無仮説は棄却される。反対に、そうでなければ、帰無仮説は棄却されなくて、構造変化はないということになる。

プログラム

```
new; cls;
load data[50,5]=d:datafile11.txt;
y=data[:,1]; x=ones(rows(data),1)~data[:,2:5];
a=1.624; @ At 5% @
call fluctest(y,x,a);

proc fluctest(y,x,a);
    local n,k,b,d,shat,e,j,ft,m,i,bj,veca;
    n=rows(x); k=cols(x);
    m=chol(x'x);
    b=inv(x'x)*x'y;
    e=y-x*b;
    shat=sqrt(e'e/(n-k));
    j=seqa(k+1,1,n-k);
    d=zeros(n-k,1);
    i=k+1;
    do while i<=n;
        bj=inv(x[1:i,.]'x[1:i,.]*)x[1:i,.]'y[1:i,];
        d[i-k,]=maxc(abs(m*(bj-b)));
        i=i+1;
    endo;
    ft=j.*d/(shat*n);
    if ft<=a;
        print "Fluctuation test (RESULT:Not reject H0)";
    else;
        print "Fluctuation test (RESULT:Reject H0)";
    endif;
    library pgraph;
    graphset;
    title ("Fluctuation TEST"); xlabel ("time j"); ylabel ("ft");
```

```

veca=ones(n-k,1)*a;
xy(j,ft~veca);
retp(ft);
endp;

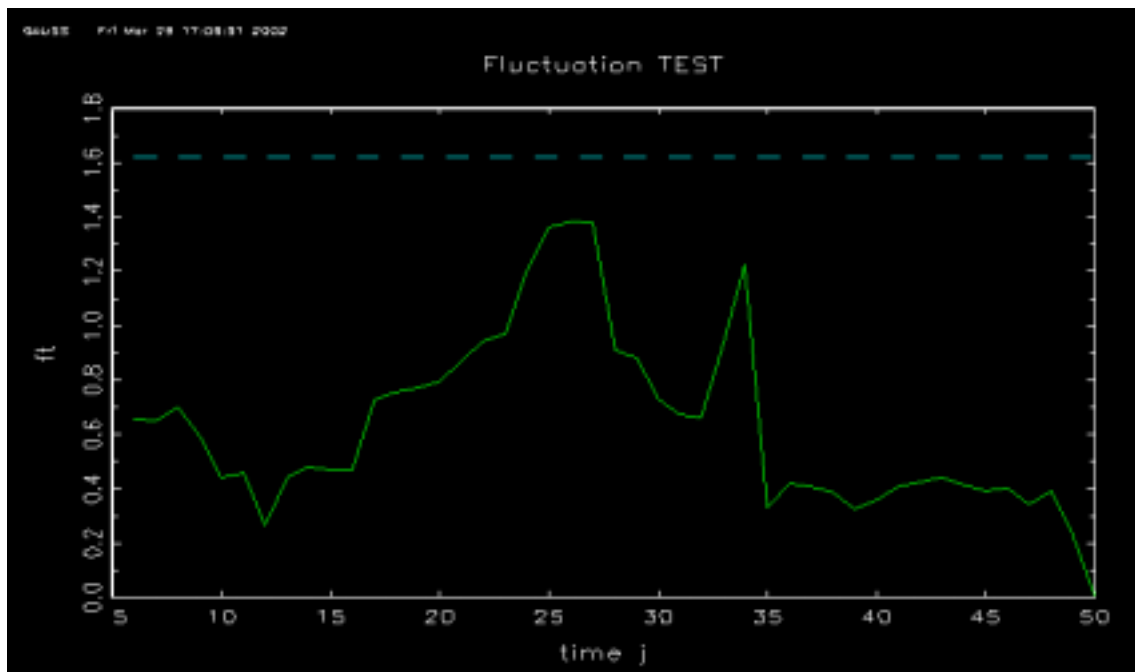
```

なお、ここで、 $\text{chol}(x'x)$ となっている計算は、Cholesky分解を求めるものである。 b_j はその時点までのデータを用いて計算された係数であって、最終的に j が n になるときは、この b_j は、全データによって計算された係数 b に一致する。したがってそのときには、この統計量は 0 になる。結果は以下のようなになる。なお、この統計量は絶対値を取っているため、最低でも 0、あらかじめ知られているそのパーセントの有意水準の上限の値のあいだに、すべての系列は構造変化がなければ位置する。構造変化が認められれば、その時点で、上限の値を超えることになる。なお、この場合には、すべての統計量の系列はグラフ的に 0 と上限のあいだにおさまっているため、このバージョンの Fluctuation Test は、構造変化がないという帰無仮説を棄却しない。すなわち、同様の 5 % の有意水準に対するその他のテスト同様に、構造変化はないと結論づけられる。

画面表示

Fluctuation test (RESULT:Not reject H0)

グラフ表示



/*

**** CUSUM Test, CUSUMSQ Test, CUSUM with OLS residuals Test,**

**** Harvey-Collier t-Test, and Fluctuation Test were all originally**

**** written in 1999 by Yosuke Amijima and his study group.**

**** (C) Copyright 1999-2003 Yosuke Amijima. All Rights Reserved.**

**** These algorithms should not be included in any other software.**

**** The copyright is strictly protected especially for these algorithms.**

**** The tests of other software may slightly differ in some points.**

***/**