'Gregory-Hansen Cointegration Test

'Reference: Gregory, A. W. and Hansen, B. E. (1996). "Residual-Based Tests for Cointegration in Models with Regime Shifts", Journal of Econometrics, Vol. 70, pp. 99-126.

group independents

independents.add x1, x2, x3

call greghansen(y,independents,3,"sic",1)

' ----------------------------------------------------------------------------------------------------

' Arguments

'-----------------------------------------------------------------------------------------------------

'series Y ' dependent variable

'group G ' group of independent variable(s) (including single series)

'scalar Model ' 2 = Level Shift, 3 = Level Shift with Trend, 4 = Regime Shift

'scalar Maxlag ' Maximum number of lags for unit root testing

'string %Criterion ' Selection criteria for unit root testing (i.e. aic / sic / hqc)

' ----------------------------------------------------------------------------------------------------

subroutine greghansen(series Y, group G, scalar Model, string %Criterion, scalar Maxlag)

smpl @all

!trim = 0.15

!maxlag = Maxlag

!n = @obs(y)

!nindep = G.@count

!lower = @round(@obs(Y)\*!trim)

!upper = @round(@obs(Y)\*(1-!trim))

matrix(!upper-!lower+1,4) GHtest

equation ghc

Table GHZ

GHZ(1,1) = "THE GREGORY-HANSEN"

GHZ(2,1) = "COINTEGRATION TEST"

if Model=2 then GHZ(3,1) = "MODEL 2: Level Shift"

 else if Model =3 then GHZ(3,1) = "MODEL 3: Level Shift with Trend"

 else if Model = 4 then GHZ(3,1) = "MODEL 4: Regime Shift"

 endif

 endif

endif

GHZ(5,1) = "ADF Procedure"

GHZ(7,1) = "t-stat"

GHZ(8,1) = "Lag"

GHZ(9,1) = "Break"

GHZ(11,1) = "Phillips Procedure"

GHZ(13,1) = "Za-stat"

GHZ(14,1) = "Za-break"

GHZ(15,1) = "Zt-stat"

GHZ(16,1) = "Zt-break"

for !ref = 2 to 4

 GHZ.setwidth(!ref) 15

next

GHZ.setlines(a4:b4) +d

GHZ.setlines(a6:b6) +d

GHZ.setlines(a10:b10) +d

GHZ.setlines(a12:b12) +d

for !i = !lower to !upper

 if Model=2 then

 'MODEL 2 - C: LEVEL SHIFT MODEL

 ghc.ls Y c G (@trend>!i-2)

 ghc.makeresid res

 uroot(adf, none, info={%criterion}, maxlag=!maxlag, save=level) res

 GHtest(!i-!lower+1,1) = level(3,1)

 GHtest(!i-!lower+1,2) = level(2,1)

 call phillips(res)

 GHtest(!i-!lower+1,3) = !Za

 GHtest(!i-!lower+1,4) = !Zt

 else if Model=3 then

 'MODEL 3 - C/T: LEVEL SHIFT WITH TREND MODEL

 ghc.ls Y c @trend G (@trend>!i-2)

 ghc.makeresid res

 uroot(adf, none, info={%criterion}, maxlag=!maxlag, save=level) res

 GHtest(!i-!lower+1,1) = level(3,1)

 GHtest(!i-!lower+1,2) = level(2,1)

 call phillips(res)

 GHtest(!i-!lower+1,3) = !Za

 GHtest(!i-!lower+1,4) = !Zt

 else if Model = 4 then

 'MODEL 4 - C/S: REGIME SHIFT MODEL

 for !g = 1 to !nindep

 G.add (@trend>!i-2)\*G(!g)

 next

 ghc.ls Y c (@trend>!i-2) G

 ghc.makeresid res

 uroot(adf, none, info={%criterion}, maxlag=!maxlag, save=level) res

 GHtest(!i-!lower+1,1) = level(3,1)

 GHtest(!i-!lower+1,2) = level(2,1)

 call phillips(res)

 GHtest(!i-!lower+1,3) = !Za

 GHtest(!i-!lower+1,4) = !Zt

 for !g = G.@count to !nindep+1 step -1

 %name = G.@seriesname(!g)

 G.drop {%name}

 next

 endif

 endif

 endif

next

 vector min\_t\_lag = @cmin(GHtest)

 vector break = @cimin(GHtest)

 GHZ(7,2) = min\_t\_lag(1)

 GHZ(8,2) = GHtest(break(1),2)

 GHZ(13,2) = min\_t\_lag(3)

 GHZ(15,2) = min\_t\_lag(4)

 if @datestr(@now,"F") = "?" then

 GHZ(9,2) = break(1) + !lower - 2

 GHZ(14,2) = break(3) + !lower - 2

 GHZ(16,2) = break(4) + !lower - 2

 else

 GHZ(9,2) = @otod(break(1) + !lower - 2)

 GHZ(14,2) = @otod(break(3) + !lower - 2)

 GHZ(16,2) = @otod(break(4) + !lower - 2)

 endif

 show GHZ

delete res level GHtest break min\_t\_lag

endsub

subroutine phillips(series y) 'MATLAB code of this routine is available at Bruce E. Hansen's website: http://www.ssc.wisc.edu/~bhansen/progs/joe\_96.html

!n = @obs(y)

equation eq1.ls y y(-1)

!be = eq1.@coefs(1)

series ue = y - !be\*y(-1)

'Bandwidth selection

!nu = @obs(ue)

equation eq2.ls ue ue(-1)

!bu = eq2.@coefs(1)

series uu = ue - !bu\*ue(-1)

!su = @sumsq(uu)/@obs(uu)

!a2 = (4\*!bu^2\*!su/(1-!bu)^8)/(!su/(1-!bu)^4)

!bw =1.3221\*((!a2\*!nu)^0.2)

!pi = @acos(-1)

!j=1

!lemda = 0

 while !j <= !bw

 series temp = ue\*ue(-!j)

 !gama = @sum(temp)/!nu

 !w=(75/(6\*!pi\*!j/!bw)^2)\*(@sin(1.2\*!pi\*!j/!bw)/(1.2\*!pi\*!j/!bw)-@cos(1.2\*!pi\*!j/!bw))

 !lemda=!lemda+!w\*!gama

 !j=!j+1

 wend

series temp = y\*y(-1) - !lemda

!p = @sum(temp)/@sumsq(y(-1))

!Za = !n\*(!p-1)

!Zt = (!p-1)/@sqrt((2\*!lemda + @sumsq(ue)/!nu)/(@sumsq(y(-1))))

smpl @all

delete eq1 eq2 ue uu temp

endsub