Applied Econometrics for Macroeconomics

Freddy Rojas Cama

Fall 2011

Assignment 2, due date: October 15th

In this assignment we determine and estimate the appropiate AR model and we undertake structural-change tests. Additionally, we discuss the nominal and empirical size of those tests. First, we donwload the quarterly real Gross domestic Product (GDP) the US for the period 1960Q1 until 2009Q4 from FRED[1] website. You can check a complete description of the original (level) serie in the appendix. Below a detail of the statistics of this GDP serie

Table 1: Statistics
(log difference of original serie)

Stats	Value	Stats	Value
Mean	0.7635	median	0.7724
Variance	0.7967	90% pctile	1.8932
Max	3.8585	95% pctile	2.2193
Min	-2.3276	99% pctile	2.6230

1. In this item we determine and then estimate the best $AR(p)$ model for quarterly real GDP growth for the US in the period 1960Q1 until 2009Q4. We estimate the following model

$$y_t = \alpha_0 + \sum_{j}^{p} \phi_j y_{t-j} + \zeta_t$$

being ϕ_j the coefficient related to lag j of the serie and ζ is an i.i.d shock. We use Freddy Rojas's MATLAB code (please see appendix for details, all the codes are in my web page[2]) We adjust the sample size in order to perform the calculations of information criteria and thus we make them comparable each other. In the following table we show the procedure in order to find the optimal lag -which minimizes the information criteria-; We start from different lags (l) and we run l regressions for each information criterion, then we calculate those criteria. As we see in the table 2 the results are the same in the case of Hannan-quinn (hq) and akaike (aic) information criteria The bayesian criterion suggest to choose either an AR(1) or AR(2).

Table 2: AR selection
(by information criteria)

Information criteria (I.C)	max lags (l) 5	10	20
`hq`	2	2	2
`bic`	1	2	1
`aic`	2	2	2
`hq-modified`	2	2	2
`bic-modified`	1	2	1
`aic-modified`	2	2	2

This time we consider to have a large model in order to have more probability of having consistent estimates -obviously we are going to lose some eficiency if the true model is an AR(1). In table 3

[1] Web page of St. Louis federal reserve.
[2] `eden.rutgers.edu\~fcama\fcama.html`

we show the results of the estimation by maximum likelihood (MLE).

Table 3: AR estimation
(log difference of original serie)

	Starting values	Coefficients	S.d	P-value
c	0.0000	0.409468	0.104895	0.0001
ϕ_1	0.0000	0.300260	0.080588	0.0003
ϕ_2	0.0000	0.165003	0.070896	0.0210

Note: HAC standard errors are estimated.

So, we conclude that we have an AR(2) for this serie.

2. By using the structural break tests discussed in class, we test for a structural break (in either the parameters or the variance of the errors) in 1990Q1. We have the following calculations in order to construc the statistics $\lambda_{ss}, \lambda_{bp}$ and λ_{cf}

Table 4: Statistics in the time-break setting up
Break in 1990Q1 (T_B)

Statistic	Value	Statistic	Value
σ^2	0.66233	$\sigma^2_{(2)}$	0.34818
$\sigma^2_{(1,2)}$	1.23094	T	198
$\sigma^2_{1,2}$	0.66836	T_1	118
$\sigma^2_{(1)}$	0.85927	T_2	77

In the table 5 we show the structural-change chow statistics, p-values are based on asymptotic distributions.

Table 5: Critical values

Break point	Test	stats	p-value
1990:1	λ_{ss}	3.20641	0.36088
1990:1	λ_{bp}	139.64948	0.00000
1990:1	λ_{cf}	0.42172	1.00000

The sample-split test (λ_{ss}) checks the null hypothesis that the AR coefficients and deterministic terms do not change during the sample period, whereas the break-point test (λ_{bp}) checks in addition the constancy of the white noise variance. In the case of chow forecast test (λ_{cf}) , this test checks whether forecasts from the model fitted to the first subsample are compatible with the observations in the second subsample. So, as we see in table 5, we cannot reject the null hypothesis of no break in parameters at this date at the 5% level. Also, we cannot reject the null hypothesis of the coefficients in model 1 are useful for doing forecast in outsample at the 5% level. However, if we additionally test constancy in the variance we may reject the null hypothesis of constancy in parameters and variance at 5% level (even in 1%). We conclude that by using asymptotic distributions we have evidence of change in variance.

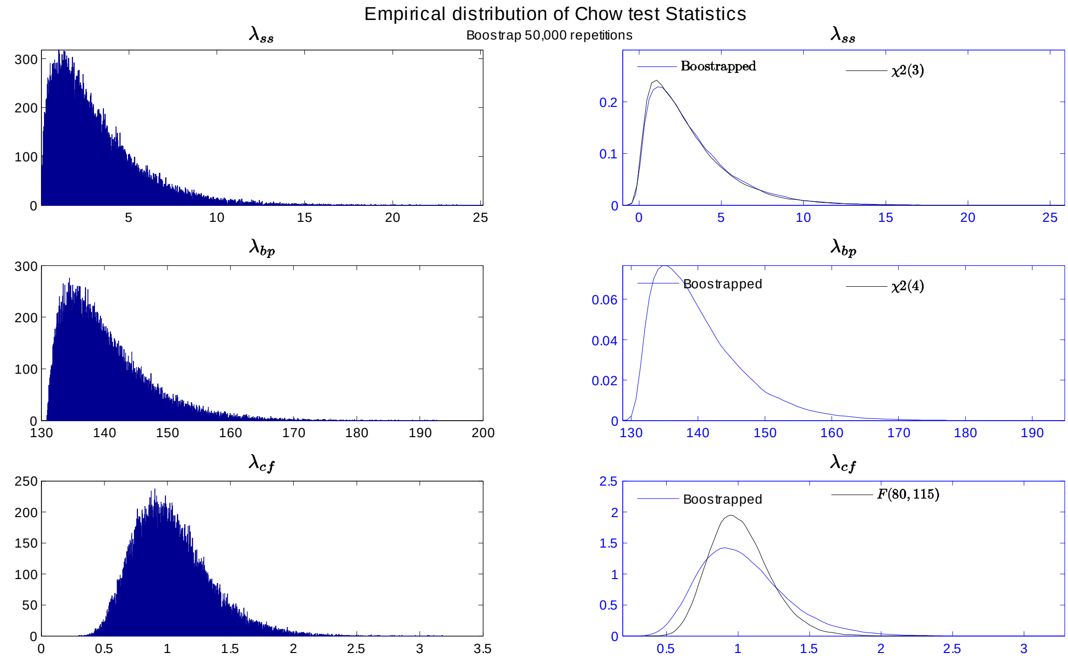
3. We compare the bootstrapped distribution of test statistics to the asymptotic distribution of the test statistic. We use different type of bootstraps; a simple one described in Lutketpol, the well-known block boostrap and the stationary blook boostrap of Politis and Romano (1994). We have

the results in the following table

Table 6: Critical values
95% critical value

	Asymptotic	Boostrap		
		Simple	Block	Stationary
λ_{ss}	7.81473	8.14260	9.94070	9.03924
λ_{bp}	9.48773	153.38410	155.21300	154.78020
λ_{cf}	1.3970989	1.57646	1.68004	1.655840

As we see high distorsions occur in λ_{bp}. Diebold and Chen (1996) reported that result, according to authors that distorsion in the difference between nominal and empirical size is additionally exacerbated if we have autocorrelated errors[3] - see Giles and Scott (1992). In the following graph, we compare the asymptotic with empirical distribution which we got from bootstrapping techniques.



We do not find any difference between empirical distribution of λ_{ss} with the asymptotic distribution, practically they are the same. There are some differences in the empirical distribution of λ_{cf} with the asymptotic distribution, we may use the aysmptotic distribution for an "*almost-good* "approximation in applied field. But we have a stunning result in the case of λ_{bp}; we have the entire distribution at the right of the asymptotic χ^2 (4) distribution. In this case we reject 100% of the time the null hypothesis. In empirical terms, Lutketpol suggests to take the critical values of the boostrap realization. But, we are a kind of skeptical about this and we are going to evaluate the power. We perform a Montecarlo Analysis for an model AR(1) and we calculate λ_{bp}; then we report the percentage of rejection of each Montecarlo draw; the steps are described in Diebold and

[3]Diebold and Chen (1996) calculate distorsions sizes around 50 and 60%.

Chen (1996), we have the amazing results:

Table 7: Rejection rate of Structural Change
Montecarlo Analysis of bootstrap technique
Diebold and Cheng (1996)

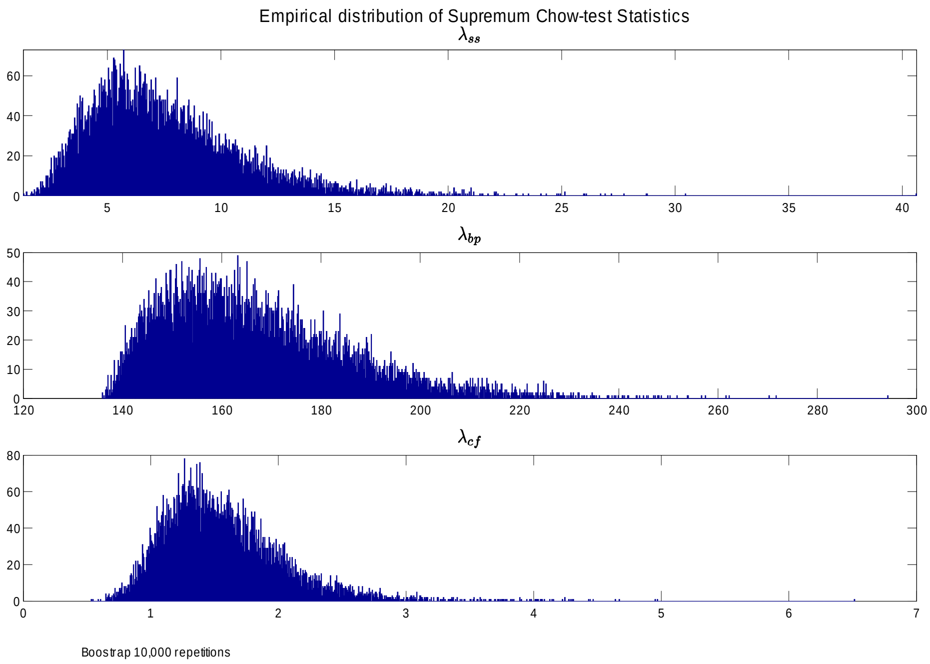
Sample	λ_{ss}	λ_{bp}	λ_{cf}
10	0.0000	0.0010	0.0010
50	0.0010	0.0020	0.0020
100	0.0030	0.0020	0.0020
1000	0.0450	0.0380	0.0490

As the sample increases the empirical size converges to the nominal size; that result is remarkable (see details in the appendix). After this results we finish with the table 8 which compare the statistics we got in item 2 with the 95% critical values -which comes from the simple boostrap draws, we state the conclusions in the same table

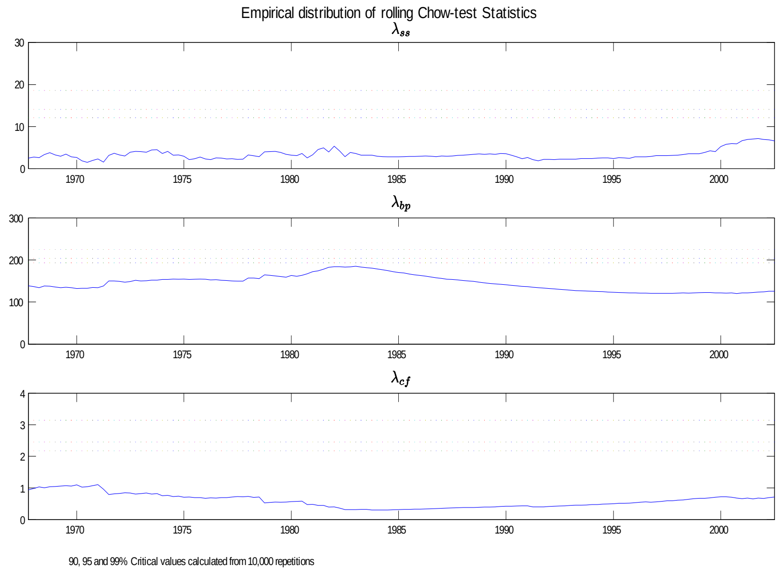
Table 8: Conclusion

Test	value	95% critical value	Conclusion
λ_{ss}	3.20641	8.14260	We cannot reject Ho
λ_{bp}	139.64948	153.38410	We cannot reject Ho
λ_{cf}	0.42172	1.57646	We cannot reject Ho

4. Now we test for parameter stability under the assumption of one break at an unknown date. We consider in any case the search of this break point between $[0.15T \;\; 0.75T]$. We draw the empirical distribution by using boostrap of this *supremum* in the following graph



We use the 90, 95 and 99% bootstrap critical values for evaluating the Ho of stability in our AR(2) process. We plot the rolling structural-change statistics



In the following table we have the critical values from the boostrap and the supremum of each test λ_{ss}, λ_{bp} and λ_{cf}

Table 1: Conclusion
Boostrap of SUP - 10000 draws

Statistic	max value (sup)	Possible break	Critical value 90%	95%	99%
λ_{ss}	7.0914	2002:2	11.86662	15.97758	24.52662
λ_{bp}	185.1946	1983:3	195.19730	208.02706	227.28503
λ_{cf}	1.1057	1971.3	2.13497	2.41725	3.35756

We conclude that there is no evidence in sample of having a break because the statistics are less than their critical values.

References

[1] James Hamilton (1995). Time Series Analysis. Princeton.

[2] Lutketpol (2002). Applied Time series Econometrics.

[3] Francis Diebold and Chen (1996). Testing structural stability with endogenous breakpoint; A size of comaprison of analytic and bootstrap procedures. Journal of Economertics 70 (1996).

[4] David Giles and Murray Scott (1992). Some consequences of using the Chow test in the context of autocorrelated disturbances. Economics Letters38.

[5] Politis & Romano (1994) "The Stationary Bootstrap" Journal of The American Statistical Association Vol. 89 N 428 Dec 1994 pp. 1303-1313

1 Appendix 1

1.1 Serie information

Title:	Real Gross Domestic Product, 3 Decimal
Series ID:	GDPC96
Source:	U.S. Department of Commerce: Bureau of Economic Analysis
Release:	Gross Domestic Product
Seasonal Adjustment:	Seasonally Adjusted Annual Rate
Frequency:	Quarterly
Units:	Billions of Chained 2005 Dollars
Date Range:	1947-01-01 to 2011-04-01
Last Updated:	2011-09-29 11:01 AM CDT
Notes:	A Guide to the National Income and Product Accounts of the United States (NIPA) - (http://www.bea.gov/national/pdf/nipaguid.pdf)

1.2 Bootstrap critical values

Table A1: Critical values
90%

	Asymptotic	Boostrap		
		Simple	Block	Stationary
λ_{ss}	6.25138	6.49453	8.07865	7.334487
λ_{bp}	7.77944	149.42140	150.55190	149.581500
λ_{cf}	1.29735	1.43127	1.51110	1.479000

Table A2: Critical values
99%

	Asymptotic	Boostrap		
		Simple	Block	Stationary
λ_{ss}	11.34486	11.79236	13.71059	12.415608
λ_{bp}	13.27670	162.61953	169.27890	165.861840
λ_{cf}	1.60498	1.92328	2.12600	2.105780

1.3 MATLAB ARMA CODE (written by me)

I use `MATLAB` for this and I wrote the entire code. The procedure is the following

1. (1) `function slog_lkhood=llog(sv0)`
   (2) `global x y p q;`
   (3) `cons=sv0(1);`
   (4) `eq=zeros(q,1);`
   (5) `if (p~=0 && q~=0)`
   (6) `ar_betas=sv0(2:p+1,1);`
   (7) `ma_betas=sv0(p+2:p+q+1,1);`
   (8) `end`
   (9) `if (p==0 && q~=0)`

```
(10) ar_betas=[];
(11) ma_betas=sv0(p+2:p+q+1,1);
(12) end
(13) if (p~=0 && q==0)
(14) ar_betas=sv0(2:p+1,1);
(15) ma_betas=0;
(16) eq=0;
(17) end
(18) if (p==0 && q==0)
(19) ar_betas=[];
(20) ma_betas=0;
(21) eq=0;
(22) end
(23) lkhoodv=[];
(24) sig=1;
(25) ys=y(p+1:size(y,1),1);
(26) for i=1:1:size(x,1);

(27)     e=ys(i)-x(i,:)*[cons; ar_betas]-eq'*ma_betas;
(28)     lkhood=1/(2^0.5*pi^0.5*sig)*exp(-(e)^2/(2*sig^2));
(29)     lkhoodv=[lkhoodv; lkhood];
(30)         if q~=0
(31)         eq=[e; eq(1:size(eq,1)-1,1)];
(32)         else
(33)         eq=0;
(34)         end
(35) end

(36) log_lkhood=log(lkhoodv);
(37) slog_lkhood=-sum(log_lkhood');
(38) end
```

Lines 27 and 28 describes the log-likelihood function, the return of this process is the sum of the log-likelihood for each observation (lines 36 and 37). I use the `MATLAB` command `fminsearch` in the following syntax

$$\texttt{[betas,fval,exitflag]=fminsearch(@llog,betas0,Options)} \tag{1}$$

The command minimize the function called `llog`. The complete MATLAB code is available in my web page[4].

[4] `http://eden.rutgers.edu/~fcama/fcama.html`

1.4 Best AR(p) - code written by me

```
clear all;
   clc;
   format short;
   global p q y upT;
   cd('C:\Users\Freddy\freddy_Backup\Freddy\Persona\Rutgers 2011\Semester III\Applied_macr
data\');

   data=xlsread('chainGDPlevel_FRED1960_2009.xls');
   ydate=data(:,1);
   y=data(:,2);
   ly=lagmatrix(y,1);
   y=(log(y(2:size(y),:))-log(ly(2:size(y),:)))*100;
   %STATS
   cd('C:\Users\Freddy\freddy_Backup\Freddy\Persona\Rutgers 2011\Semester III\Applied_macr
   stats_serie=[mean(y); var(y); median(y); max(y); min(y); prctile(y,90); prctile(y,95);
prctile(y,99)];
   save('stats_gdpr_q','stats_serie','/ASCII');
   m='conditional';
   upT=2;
   aic_v=[];
   bic_v=[];
   hq_v=[];
   betas_v=[];
   flag_v=[];
   bsv=[0.5];
   for j=1:1:upT;
   p=j;
   q=0;
   bsv=[bsv(1:p,1); 0];
   cd('C:\Users\Freddy\freddy_Backup\Freddy\Persona\Rutgers 2011\Semester III\Applied_macr
   [betas,flag,fval,aic,bic,hq,ssr,e]=arma(y,p,q,m,bsv,upT);
   bsv=betas;
   aic_v=[aic_v; aic];
   bic_v=[bic_v; bic];
   hq_v=[hq_v; hq];
   betas_v=[betas_v; [betas' zeros(1,upT-j)]];
   flag_v=[flag_v; flag];
   end
   [C_bic,I_bic]=min(bic_v);
   [C_hq,I_hq]=min(hq_v);
   [C_aic,I_aic]=min(aic_v);
   cd('C:\Users\Freddy\freddy_Backup\Freddy\Persona\Rutgers 2011\Semester III\Applied_macr
   save(sprintf('I_bic_%d',upT),'I_bic','/ASCII');
   save(sprintf('I_hq_%d',upT),'I_hq','/ASCII');
   save(sprintf('I_aic_%d',upT),'I_aic','/ASCII');
```

1.5 How I construct the statistics - MATLAB function break_ar.m

```
function [stat, pvalue]=break_ar(e1,e2,ep,efull,k1,k2,k)
   % e1 ; errors from sample-1 estimation
```

```
% e2 ; errors from sample-2 estimation
% efull ; errors from full-sample estimation
% ep ; [efull1 efull2] efullj from errors from full-sample estimation but
% adjusted in the sample of ej-sample
% k1 ; number of parameters in sample-1 estimation
% k2 ; number of parameters in sample-2 estimation
% k ; number of parameters in full-sample estimation
% Last update 11/10/2011
% Made by Freddy Rojas Cama
if size(e1,1)+size(e2,1)~=size(ep,1);
 error('vector of errors from split sample needs to add up the size of the
vector of errors from the full sample - adjusted by AR estimation')
end
T1=size(e1,1);
T2=size(e2,1);
T=size(efull,1);
sgm2=sum(ep.^2)/(T1+T2);
sgm2_alt=sum(ep(1:size(e1,1),:).^2)/(T1)+sum(ep(size(e1,1)+1:size(ep,1),:).^2)./(T2);
sgm2_full=sum(efull.^2)/T;
sgm1_2=sum(e1.^2)./(T1);
sgm2_2=sum(e2.^2)./(T2);
%[sgm1_2 sgm2_2 sgm2_alt sgm2_full sgm2]
kdf=k1+k2-k;
lbda=(T1+T2)*(log(sgm2)-log(1/(T1+T2)*(T1*sgm1_2+T2*sgm2_2)));
lbda_bp=(T1+T2)*log(sgm2_alt)-T1*log(sgm1_2)-T2*log(sgm2_2);
lbda_cf=((T)*sgm2_full-T1*sgm1_2)/(T1*sgm1_2)*(T1-kdf)/(T-T1);
stat=[lbda; lbda_bp; lbda_cf];
pvalue=[1-chi2cdf(lbda,kdf); 1-chi2cdf(lbda_bp,k+1); 1-fcdf(lbda_cf,T-T1,T1-kdf)];
end
```

1.6 Boostrap in order to get the empirical distribution N=50000

```
clear all;
clc;
format short;
cd('C:\Users\Freddy\freddy_Backup\Freddy\Persona\Rutgers 2011\Semester III\Applied_macr
data\');

data=xlsread('chainGDPlevel_FRED1960_2009.xls');
ydate=data(:,1);
y=data(:,2);
ly=lagmatrix(y,1);
y=(log(y(2:size(y),:))-log(ly(2:size(y),:)))*100;
p=2;
tb=121;
ndel=1;
ydata=[];
for j=1:1:p
 ylag=lagmatrix(y,j);
 ydata=[ydata ylag];
end
```

```
xdata=ydata(p+1:size(ydata,1),:);
xdata=[ones(size(xdata,1),1) xdata];
ys=y(p+1:size(y),1);
betas_ols=inv(xdata'*xdata)*xdata'*ys;
e0=ys-xdata*betas_ols;
stat_boots=[];
nboost=50000;
for w=1:1:nboost;
 display('boring boostrapping:  Iterations');
 clc; w
cd('C:\Users\Freddy\freddy_Backup\Freddy\Persona\Rutgers 2011\Semester III\Applied_macr

stateq=ceil(rand*(100));
[e_sam]=resampling(e0,size(e0,1)+p);
e_sam_st=e_sam-sum(e_sam)/size(e_sam,1);
y0=betas_ols(1)/(1-betas_ols(2)-betas_ols(3));
ysampl=zeros(size(e_sam_st,1),1);
y0v=[y0 y0];
for i=1:1:size(e_sam_st,1);
 ysampl(i)=[1 y0v]*[betas_ols(1); betas_ols(2:p+1,1)]+e_sam_st(i);
 y0v=[ysampl(i) y0v(1, 1:size(y0v,2)-1)];
end
ydatab=[]; yb=ysampl;
for j=1:1:p
 ylag=lagmatrix(yb,j);
 ydatab=[ydatab ylag];
end
xdatab=ydatab(p+1:size(ydatab,1),:);
xdatab=[ones(size(xdatab,1),1) xdatab];
ysb=yb(p+1:size(yb),1);
bts_ols_b=inv(xdatab'*xdatab)*xdatab'*ysb;
e0b=ysb-xdatab*bts_ols_b;
e0bp=[e0b(1:tb-p-ndel,:); e0b(tb-p+p+ndel:size(e0b,1),:)];
%model1
xdatab1=xdatab(1:tb-p-ndel,:);
ysb1=ysb(1:tb-p-ndel,:);
bts_ols_b1=inv(xdatab1'*xdatab1)*xdatab1'*ysb1;
e1b=ysb1-xdatab1*bts_ols_b1;
%model2
xdatab2=xdatab(tb-p+p+ndel:size(xdatab,1),:);
ysb2=ysb(tb-p+p+ndel:size(xdatab,1),:);
bts_ols_b2=inv(xdatab2'*xdatab2)*xdatab2'*ysb2;
e2b=ysb2-xdatab2*bts_ols_b2;
cd('C:\Users\Freddy\freddy_Backup\Freddy\Persona\Rutgers 2011\Semester III\Applied_macr
[stat, pvalue]=break_ar(e1b,e2b,e0bp,e0b,size(bts_ols_b1,1),size(bts_ols_b2,1),size(bts
stat_boots=[stat_boots; stat'];
end
s1=sort(stat_boots(:,1));
s2=sort(stat_boots(:,2));
s3=sort(stat_boots(:,3));
```

```
stats_99=[s1(round(0.99*size(s1,1))) s2(round(0.99*size(s2,1))) s3(round(0.99*size(s3,
;
stats_95=[s1(round(0.95*size(s1,1))) s2(round(0.95*size(s2,1))) s3(round(0.95*size(s3,
;
stats_90=[s1(round(0.90*size(s1,1))) s2(round(0.90*size(s2,1))) s3(round(0.90*size(s3,
cd('C:\Users\Freddy\freddy_Backup\Freddy\Persona\Rutgers 2011\Semester III\Applied_macro
%Saving results in TXT
save('stat_boots','stat_boots','/ASCII');
save('e0_confirmate','e0','/ASCII');
save('stats_99','stats_99','/ASCII');
save('stats_95','stats_95','/ASCII');
save('stats_90','stats_90','/ASCII');
cd('C:\Users\Freddy\freddy_Backup\Freddy\Persona\Rutgers 2011\Semester III\Applied_macro
%Saving in matlab format
save('stat_boots.mat','stat_boots');
save('stats_99.mat','stats_99');
save('stats_95.mat','stats_95');
save('stats_90.mat','stats_90');
%GRAPHS
figure3 = figure('Color',[1 1 1]);
%suptitle('Empirical distribution of Chow test Statistics');
%suptitle('Boostrap 50,000 repetitions');
subplot(3,2,1);
hist(stat_boots(:,1),1000);
title('$\lambda_{ss}$','fontsize',14,'interpreter','latex');
axis tight
subplot(3,2,3);
hist(stat_boots(:,2),1000);
title('$\lambda_{bp}$','fontsize',14,'interpreter','latex');
subplot(3,2,5);
hist(stat_boots(:,3),1000);
title('$\lambda_{cf}$','fontsize',14,'interpreter','latex');
axes1 = axes('Visible','off','Parent',figure3,'Tag','suptitle',...
 'Position',[0 1 1 1]);
text('Parent',axes1,...
 'String','Empirical distribution of Chow test Statistics',...
 'Position',[0.515797788309637 -0.0237020316027088 9.16025403784439],...
 'HorizontalAlignment','center',...
 'FontSize',14);
text('Parent',axes1,...
 'String','Boostrap 50,000 repetitions',...
 'Position',[0.515797788309637 -0.0537020316027088 9.16025403784439],...
 'HorizontalAlignment','center',...
 'FontSize',10);
cd('C:\Users\Freddy\freddy_Backup\Freddy\Persona\Rutgers 2011\Semester III\Applied_macro
saveas(gcf,'graph.eps')
saveas(gcf,'graph.png')
%GRAPHS
%figure4 = figure('Color',[1 1 1]);
%suptitle('Empirical distribution of Chow test Statistics');
%suptitle('Boostrap 50,000 repetitions');
```

```
subplot(3,2,2);
[f1,xi1]=ksdensity(stat_boots(:,1));
ksdensity(stat_boots(:,1))
title('$\lambda_{ss}$','fontsize',14,'interpreter','latex');
axis([-1 max(xi1) 0 0.3])
h = legend('Boostrapped',2);
set(h,'Interpreter','latex')
legend('boxoff')
ax1 = gca;
set(ax1,'XColor','b','YColor','b')
ax2 = axes('Position',get(ax1,'Position'),...
 'XAxisLocation','top',...
 'YAxisLocation','right',...
 'Color','none',...
 'XColor','k','YColor','k');
x2 = chi2rnd(3,nboost,1);
[f2,xi2]=ksdensity(x2);
axis off
hl2 = line(xi2,f2,'Color','k','Parent',ax2);
axis([-1 max(xi1) 0 0.3])
h = legend('$\chi{2}$(3)');
set(h,'Interpreter','latex','Position',[0.7 0.87 0.14 0.0526711813393529]);
legend('boxoff')
subplot(3,2,4);
[f1,xi1] = ksdensity(stat_boots(:,2));
ksdensity(stat_boots(:,2))
title('$\lambda_{bp}$','fontsize',14,'interpreter','latex');
axis([min(xi1) max(xi1) 0 max(f1)])
h = legend('Boostrapped',2);
set(h,'Interpreter','none')
legend('boxoff')
ax1 = gca;
set(ax1,'XColor','b','YColor','b')
ax2 = axes('Position',get(ax1,'Position'),...
 'XAxisLocation','top',...
 'YAxisLocation','right',...
 'Color','none',...
 'XColor','k','YColor','k');
x2 = chi2rnd(3,nboost,1);
[f2,xi2]=ksdensity(x2);
axis off
hl2 = line(xi2,f2,'Color','k','Parent',ax2);
axis([min(xi1) max(xi1) 0 max(f1)])
h = legend('$\chi{2}(4)$');
set(h,'Interpreter','latex','Position',[0.7 0.57 0.14 0.0526711813393529]);
legend('boxoff')
subplot(3,2,6);
[f1,xi1] = ksdensity(stat_boots(:,3));
ksdensity(stat_boots(:,3));
title('$\lambda_{cf}$','fontsize',14,'interpreter','latex');
axis([min(xi1) max(xi1) 0 2.5])
```

```
h = legend('Boostrapped',2);
set(h,'Interpreter','none')
legend('boxoff')
ax1 = gca;
set(ax1,'XColor','b','YColor','b')
ax2 = axes('Position',get(ax1,'Position'),...
 'XAxisLocation','top',...
 'YAxisLocation','right',...
 'Color','none',...
 'XColor','k','YColor','k');
x2 = frnd(198-118,118-3,nboost,1);
[f2,xi2]=ksdensity(x2);
axis off
hl2 = line(xi2,f2,'Color','k','Parent',ax2);
axis([min(xi1) max(xi1) 0 2.5])
h = legend('$F(80,115)$');
set(h,'Interpreter','latex','Position',[0.7 0.28 0.14 0.0526711813393529]);
legend('boxoff')
axes1 = axes('Visible','off','Parent',figure3,'Tag','suptitle',...
 'Position',[0 1 1 1]);
saveas(gcf,'graph_kden_overlap.eps')
saveas(gcf,'graph_kden_overlap.png')
saveas(gcf,'graph_kden_overlap.fig')
% Create text
%text('Parent',axes1,'String','Boostrap 50,000 repetitions',...
 % 'Position',[0.204581358609795 -0.696388261851016 9.16025403784439],...
 %'HorizontalAlignment','center');
```

1.7 Boostrap in order to get the empirical distribution of the supremum N=10000

```
clear all;
clc;
format short;
cd('C:\Users\Freddy\freddy_Backup\Freddy\Persona\Rutgers 2011\Semester III\Applied_macro
data\');

data=xlsread('chainGDPlevel_FRED1960_2009.xls');
ydate=data(:,1);
y=data(:,2);
ly=lagmatrix(y,1);
y=(log(y(2:size(y),:))-log(ly(2:size(y),:)))*100;
p=2;
%tb=121;
ndel=1;
ydata=[];
for j=1:1:p
 ylag=lagmatrix(y,j);
 ydata=[ydata ylag];
end
xdata=ydata(p+1:size(ydata,1),:);
```

```
xdata=[ones(size(xdata,1),1) xdata];
ys=y(p+1:size(y),1);
betas_ols=inv(xdata'*xdata)*xdata'*ys;
e0=ys-xdata*betas_ols;
stat_boots_sup=[];
tb_l=30;
tb_u=170;
nboost=10000;
for w=1:1:nboost;
 display('boring boostrapping:  Iterations');
 clc; w
cd('C:\Users\Freddy\freddy_Backup\Freddy\Persona\Rutgers 2011\Semester III\Applied_macr

[e_sam]=resampling(e0,size(e0,1)+p);
e_sam_st=e_sam-sum(e_sam)/size(e_sam,1);
y0=betas_ols(1)/(1-betas_ols(2)-betas_ols(3));
ysampl=zeros(size(e_sam_st,1),1);
y0v=[y0 y0];
for i=1:1:size(e_sam_st,1);
 ysampl(i)=[1 y0v]*[betas_ols(1); betas_ols(2:p+1,1)]+e_sam_st(i);
 y0v=[ysampl(i) y0v(1, 1:size(y0v,2)-1)];
end
ydatab=[]; yb=ysampl;
for j=1:1:p
 ylag=lagmatrix(yb,j);
 ydatab=[ydatab ylag];
end
xdatab=ydatab(p+1:size(ydatab,1),:);
xdatab=[ones(size(xdatab,1),1) xdatab];
ysb=yb(p+1:size(yb),1);
bts_ols_b=inv(xdatab'*xdatab)*xdatab'*ysb;
e0b=ysb-xdatab*bts_ols_b;
stat_ukn=[];
 for tb=tb_l:1:tb_u;

 e0bp=[e0b(1:tb-p-ndel,:); e0b(tb-p+p+ndel:size(e0b,1),:)];
 %model1
 xdatab1=xdatab(1:tb-p-ndel,:);
 ysb1=ysb(1:tb-p-ndel,:);
 bts_ols_b1=inv(xdatab1'*xdatab1)*xdatab1'*ysb1;
 e1b=ysb1-xdatab1*bts_ols_b1;
 %model2
 xdatab2=xdatab(tb-p+p+ndel:size(xdatab,1),:);
 ysb2=ysb(tb-p+p+ndel:size(xdatab,1),:);
 bts_ols_b2=inv(xdatab2'*xdatab2)*xdatab2'*ysb2;
 e2b=ysb2-xdatab2*bts_ols_b2;
 cd('C:\Users\Freddy\freddy_Backup\Freddy\Persona\Rutgers 2011\Semester III\Applied_mac
 [stat, pvalue]=break_ar(e1b,e2b,e0bp,e0b,size(bts_ols_b1,1),size(bts_ols_b2,1),size(bt
 stat_ukn=[stat_ukn; stat'];
 end
```

```
 stat_sup=[max(stat_ukn)];
 stat_boots_sup=[stat_boots_sup; stat_sup];
end
s1=sort(stat_boots_sup(:,1));
s2=sort(stat_boots_sup(:,2));
s3=sort(stat_boots_sup(:,3));
stats_99_sup=[s1(round(0.99*size(s1,1))) s2(round(0.99*size(s2,1))) s3(round(0.99*size
;
stats_95_sup=[s1(round(0.95*size(s1,1))) s2(round(0.95*size(s2,1))) s3(round(0.95*size
;
stats_90_sup=[s1(round(0.90*size(s1,1))) s2(round(0.90*size(s2,1))) s3(round(0.90*size
cd('C:\Users\Freddy\freddy_Backup\Freddy\Persona\Rutgers 2011\Semester III\Applied_macr
%Saving results in TXT
save('stat_boots_sup','stat_boots_sup','/ASCII');
save('e0','e0','/ASCII');
save('stats_99_sup','stats_99_sup','/ASCII');
save('stats_95_sup','stats_95_sup','/ASCII');
save('stats_90_sup','stats_90_sup','/ASCII');
cd('C:\Users\Freddy\freddy_Backup\Freddy\Persona\Rutgers 2011\Semester III\Applied_macr
%Saving in matlab format
save('stat_boots_sup.mat','stat_boots_sup');
save('e0.mat','e0');
save('stats_99_sup.mat','stats_99_sup');
save('stats_95_sup.mat','stats_95_sup');
save('stats_90_sup.mat','stats_90_sup');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%GRAPHS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure3 = figure('Color',[1 1 1]);
%suptitle('Empirical distribution of Chow test Statistics');
%suptitle('Boostrap 50,000 repetitions');
subplot(3,1,1);
hist(stat_boots_sup(:,1),1000);
title('$\lambda_{ss}$','fontsize',14,'interpreter','latex');
axis tight
subplot(3,1,2);
hist(stat_boots_sup(:,2),1000);
title('$\lambda_{bp}$','fontsize',14,'interpreter','latex');
subplot(3,1,3);
hist(stat_boots_sup(:,3),1000);
title('$\lambda_{cf}$','fontsize',14,'interpreter','latex');
axes1 = axes('Visible','off','Parent',figure3,'Tag','suptitle',...
 'Position',[0 1 1 1]);
text('Parent',axes1,...
 'String','Empirical distribution of Supremum Chow-test Statistics',...
 'Position',[0.515797788309637 -0.0237020316027088 9.16025403784439],...
 'HorizontalAlignment','center',...
 'FontSize',14);
text('Parent',axes1,...
 'String','Boostrap 10,000 repetitions',...
 'Position',[0.515797788309637 -0.0537020316027088 9.16025403784439],...
```

```
 'HorizontalAlignment','center',...
 'FontSize',10);
%save
cd('C:\Users\Freddy\freddy_Backup\Freddy\Persona\Rutgers 2011\Semester III\Applied_macr
saveas(gcf,'SUP_graph.eps')
saveas(gcf,'SUP_graph.png')
saveas(gcf,'SUP_graph.fig')
```

1.8 Block boostrap STATA ado code (written by me)

```
************************************
* Block Boostrap
************************************
* Programmed by Freddy Rojas C.
* Universidad de Chile & Rutgers University
* freddyr@iadb.org
* last update October 15, 2011
mata:
mata clear
real matrix block(real matrix m, t)
{
nr=rows(m)
s1=ceil(nr^t)
tr=floor(nr/s1)
cv=0
V=J(nr-s1*tr,1,.)
//s1
//tr
//V
if (nr != s1*tr) {
count=round(uniform(1,1)*(nr-s1+0.99)+0.5)
V[1..nr-s1*tr,1]=m[count..count+rows(V)-1,1]
}
if (nr==s1*tr) {
cv=1
ntr=tr
}
else {
cv=0
ntr=tr+1
}
B=J(nr,1,.)
ix0=1
for (i = 1; i <= ntr; i++) {
count=round(uniform(1,1)*(nr-s1+0.99)+0.5)
//count
if (ix0<=nr-s1+1+cv) {
B[ix0..s1+ix0-1,1]=m[count..count+s1-1,1]
ix0=s1+ix0
}
else {
```

```
B[ix0..rows(V)+ix0-1,1]=V
ix0=rows(V)+ix0
}
}
B=B[1..nr,1]
return(B)
}
mata mosave block(), replace
end
**********************************************************************
capture program drop blockboots
program define blockboots
version 10
syntax varlist(max=1) [, size(real 3)]
display in ye " This program performs Block Bootstrapping "
display in ye " "
display in ye " Programmed by Freddy Rojas C."
display in ye " Universidad de Chile "
display in ye " freddyr@iadb.org "
display in gr " last update april 28th 2008"
display " "
mkmat `varlist', mat(`varlist'_mat)
mata:  st_matrix("resid_boots",block(st_matrix("`varlist'_mat"),1/`size'))
end
```

1.9 Block stationarity boostrap STATA code (written by me)

```
***********************************
* Stationary Block Boostrap
***********************************
* Programmed by Freddy Rojas C.
* Reference:  Politis & Romano (1994) "The Stationary Bootstrap" Journal of
The
* American Statistical Association Vol.  89 N 428 Dec 1994 pp.  1303-1313
* Universidad de Chile
* freddyr@iadb.org
* last update april 28th 2008
mata:
mata clear
real matrix block_st(real matrix m, p)
{

   N=rows(m)
   max=round(log(0.00000001)/log(1-p)+1)
   B=J(N+max,1,.)
for (j = 1; j <= N; ) {
   i=round(uniform(1,1)*(N-1+0.999)+0.5)
   l=round(log(uniform(1,1))/log(1-p)+1)

   if (i+l>=N) {
      B[j..j+N-i,1]=m[i..N,1]
```

```
        j=j+N-i+1
        for (s=1; s<l-(N-i); s++) {
            ix=mod(N+s,N)
                if (ix==0) {
                ix=N
                }
        B[j,1]=m[ix,1]
        j=j+1
        }

    }
    else {

    B[j..j+l-1,1]=m[i..i+l-1,1]
    j=j+l
    j
    l
    }
}
B=B[1..N,1]
return(B)
}
mata mosave block_st(), replace
end
***************************************************************************
capture program drop blockbootsst
program define blockbootsst
version 10
syntax varlist(max=1) [, prob(real 0.5)]
display in ye " This program performs Stationary Block Bootstrapping "
display in ye " "
display in ye " Programmed by Freddy Rojas C."
display in ye " Universidad de Chile "
display in ye " freddyr@iadb.org "
display in gr " last update april 28th 2008"
display " "
mkmat ‘varlist’, mat(‘varlist’_mat)
mata:  st_matrix("resid_boots",block_st(st_matrix("‘varlist’_mat"),‘prob’))
end
```

1.10 MATLAB code for calculating the Andrews and Quandt procedure when the break date is unknown

```
clear all;
  clc;
  format short;
  cd('C:\Users\Freddy\freddy_Backup\Freddy\Persona\Rutgers 2011\Semester III\Applied_macr
data\');

  data=xlsread('chainGDPlevel_FRED1960_2009.xls');
  ydate=data(:,1);
```

```
y=data(:,2);
ly=lagmatrix(y,1);
y=(log(y(2:size(y),:))-log(ly(2:size(y),:)))*100;
cd('C:\Users\Freddy\freddy_Backup\Freddy\Persona\Rutgers 2011\Semester III\Applied_macr
load stats_90_sup.mat, load stats_95_sup.mat, load stats_99_sup.mat;
p=2;
ndel=1;
ydata=[];
for j=1:1:p
 ylag=lagmatrix(y,j);
 ydata=[ydata ylag];
end
xdata=ydata(p+1:size(ydata,1),:);
xdata=[ones(size(xdata,1),1) xdata];
ys=y(p+1:size(y),1);
betas_ols=inv(xdata'*xdata)*xdata'*ys;
e0=ys-xdata*betas_ols;
tb_l=31;
tb_u=170;
stat_ukn=[];
 for tb=tb_l:1:tb_u;

 e0p=[e0(1:tb-p-ndel,:); e0(tb-p+p+ndel:size(e0,1),:)];
 %model1
 xdata1=xdata(1:tb-p-ndel,:);
 ys1=ys(1:tb-p-ndel,:);
 bts_ols_1=inv(xdata1'*xdata1)*xdata1'*ys1;
 e1=ys1-xdata1*bts_ols_1;
 %model2
 xdata2=xdata(tb-p+p+ndel:size(xdata,1),:);
 ys2=ys(tb-p+p+ndel:size(xdata,1),:);
 bts_ols_2=inv(xdata2'*xdata2)*xdata2'*ys2;
 e2=ys2-xdata2*bts_ols_2;
 cd('C:\Users\Freddy\freddy_Backup\Freddy\Persona\Rutgers 2011\Semester III\Applied_mac
 [stat, pvalue]=break_ar(e1,e2,e0p,e0,size(bts_ols_1,1),size(bts_ols_2,1),size(betas_ols
 stat_ukn=[stat_ukn; stat'];

 end

 stat_sup=[max(stat_ukn)];
 [vx,ix]=max(stat_ukn);
 stat_sup_date=[stat_sup' [ydate(tb_l+p+ix(1)+1); ydate(tb_l+p+ix(2)+1); ydate(tb_l+p+i

cd('C:\Users\Freddy\freddy_Backup\Freddy\Persona\Rutgers 2011\Semester III\Applied_macr
%Saving results in TXT
save('stat_ukn','stat_ukn','/ASCII');
save('e0_ukn','e0','/ASCII');
cd('C:\Users\Freddy\freddy_Backup\Freddy\Persona\Rutgers 2011\Semester III\Applied_macr
%Saving in matlab format
save('stat_ukn.mat','stat_ukn');
save('e0_ukn.mat','e0');
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%GRAPHS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure5 = figure('Color',[1 1 1]);
%suptitle('Empirical distribution of Chow test Statistics');
%suptitle('Boostrap 50,000 repetitions');
subplot(3,1,1);
%chiz = chi2inv(0.95,size(betas_ols,1));
%line([ydate(tb_l+p:tb_u+2,:)],stat_ukn(:,1));
plot([ydate(tb_l+p:tb_u+2,:)],stat_ukn(:,1),[ydate(tb_l+p:tb_u+2,:)],stats_95_sup(1),
[ydate(tb_l+p:tb_u+2,:)],stats_90_sup(1), [ydate(tb_l+p:tb_u+2,:)],stats_99_sup(1));
axis([ydate(tb_l+p,1) ydate(tb_u+2,1) 0 30]);
title('$\lambda_{ss}$','fontsize',14,'interpreter','latex');
subplot(3,1,2);
%chiz = chi2inv(0.95,size(betas_ols,1)+1);
%plot([ydate(tb_l+p:tb_u+2,:)],stat_ukn(:,2));
plot([ydate(tb_l+p:tb_u+2,:)],stat_ukn(:,2),[ydate(tb_l+p:tb_u+2,:)],stats_95_sup(2),
'-', [ydate(tb_l+p:tb_u+2,:)],stats_90_sup(2), [ydate(tb_l+p:tb_u+2,:)],stats_99_sup(2));
axis([ydate(tb_l+p,1) ydate(tb_u+2,1) 0 300]);
title('$\lambda_{bp}$','fontsize',14,'interpreter','latex');
subplot(3,1,3);
%Fz = chi2inv(0.95,size(betas_ols,1));
%line([ydate(tb_l+p:tb_u+2,:)],stat_ukn(:,3));
plot([ydate(tb_l+p:tb_u+2,:)],stat_ukn(:,3),[ydate(tb_l+p:tb_u+2,:)],stats_95_sup(3),
[ydate(tb_l+p:tb_u+2,:)],stats_90_sup(3), [ydate(tb_l+p:tb_u+2,:)],stats_99_sup(3));
axis([ydate(tb_l+p,1) ydate(tb_u+2,1) 0 4]);
title('$\lambda_{cf}$','fontsize',14,'interpreter','latex');
axes1 = axes('Visible','off','Parent',figure5,'Tag','suptitle',...
 'Position',[0 1 1 1]);
text('Parent',axes1,...
 'String','Empirical distribution of rolling Chow-test Statistics',...
 'Position',[0.515797788309637 -0.0237020316027088 9.16025403784439],...
 'HorizontalAlignment','center',...
 'FontSize',14);
text('Parent',axes1,...
 'String','90, 95 and 99% Critical values calculated from 10,000 repetitions',...
 'Position',[0.515797788309637 -0.950037020316027088 9.16025403784439],...
 'HorizontalAlignment','center',...
 'FontSize',10);
%save
cd('C:\Users\Freddy\freddy_Backup\Freddy\Persona\Rutgers 2011\Semester III\Applied_macro
saveas(gcf,'ukn_graph.eps')
saveas(gcf,'ukn_graph.png')
```

1.11 Diebold and Cheng (1996) Montecarlo Analysis - written by me following the algorithm

```
clear all;
clc;
format short;
```

```
mc=100;
phi_ols=[0.2; 0.3; 0.4];
num_ss=0;
num_bp=0;
num_cf=0;
for s=1:1:mc;
 p=size(phi_ols,1)-1;
 y0=phi_ols(1)/(1-phi_ols(2)-phi_ols(3));
 ysampl=zeros(200,1);
 y0v=[y0 y0];
 for i=1:1:size(ysampl,1);
 ysampl(i)=[1 y0v]*[phi_ols(1); phi_ols(2:p+1,1)]+0.01*rand(1,1);
 y0v=[ysampl(i) y0v(1, 1:size(y0v,2)-1)];
 end
 y=ysampl;

 tb=121;
 ndel=1;
 ydata=[];
 for j=1:1:p
 ylag=lagmatrix(y,j);
 ydata=[ydata ylag];
 end
 xdata=ydata(p+1:size(ydata,1),:);
 xdata=[ones(size(xdata,1),1) xdata];
 ys=y(p+1:size(y),1);
 betas_ols=inv(xdata'*xdata)*xdata'*ys;
 e0=ys-xdata*betas_ols;
 e0p=[e0(1:tb-p-ndel,:); e0(tb-p+p+ndel:size(e0,1),:)];
 %model1
 xdata1=xdata(1:tb-p-ndel,:);
 ys1=ys(1:tb-p-ndel,:);
 bts_ols_1=inv(xdata1'*xdata1)*xdata1'*ys1;
 e1=ys1-xdata1*bts_ols_1;
 %model2
 xdata2=xdata(tb-p+p+ndel:size(xdata,1),:);
 ys2=ys(tb-p+p+ndel:size(xdata,1),:);
 bts_ols_2=inv(xdata2'*xdata2)*xdata2'*ys2;
 e2=ys2-xdata2*bts_ols_2;
 cd('C:\Users\Freddy\freddy_Backup\Freddy\Persona\Rutgers 2011\Semester III\Applied_mac
 [stat_0, pvalue_0]=break_ar(e1,e2,e0p,e0,size(bts_ols_1,1),size(bts_ols_2,1),size(betas
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% BOOTSTRAP
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
stat_boots=[];
nboost=1000;
for w=1:1:nboost;
 display('boring boostrapping:  Iterations');
 clc; s
cd('C:\Users\Freddy\freddy_Backup\Freddy\Persona\Rutgers 2011\Semester III\Applied_macr
```

```
[e_sam]=resampling(e0,size(e0,1)+p);
 e_sam_st=e_sam-sum(e_sam)/size(e_sam,1);
y0=betas_ols(1)/(1-betas_ols(2)-betas_ols(3));
ysampl=zeros(size(e_sam_st,1),1);
y0v=[y0 y0];
for i=1:1:size(e_sam_st,1);
 ysampl(i)=[1 y0v]*[betas_ols(1); betas_ols(2:p+1,1)]+e_sam_st(i);
 y0v=[ysampl(i) y0v(1, 1:size(y0v,2)-1)];
end
ydatab=[]; yb=ysampl;
for j=1:1:p
 ylag=lagmatrix(yb,j);
 ydatab=[ydatab ylag];
end
xdatab=ydatab(p+1:size(ydatab,1),:);
xdatab=[ones(size(xdatab,1),1) xdatab];
ysb=yb(p+1:size(yb),1);
bts_ols_b=inv(xdatab'*xdatab)*xdatab'*ysb;
e0b=ysb-xdatab*bts_ols_b;
e0bp=[e0b(1:tb-p-ndel,:); e0b(tb-p+p+ndel:size(e0b,1),:)];
%model1
xdatab1=xdatab(1:tb-p-ndel,:);
ysb1=ysb(1:tb-p-ndel,:);
bts_ols_b1=inv(xdatab1'*xdatab1)*xdatab1'*ysb1;
e1b=ysb1-xdatab1*bts_ols_b1;
%model2
xdatab2=xdatab(tb-p+p+ndel:size(xdatab,1),:);
ysb2=ysb(tb-p+p+ndel:size(xdatab,1),:);
bts_ols_b2=inv(xdatab2'*xdatab2)*xdatab2'*ysb2;
e2b=ysb2-xdatab2*bts_ols_b2;
cd('C:\Users\Freddy\freddy_Backup\Freddy\Persona\Rutgers 2011\Semester III\Applied_macr
[stat, pvalue]=break_ar(e1b,e2b,e0bp,e0b,size(bts_ols_b1,1),size(bts_ols_b2,1),size(bts
stat_boots=[stat_boots; stat'];
end
s1=sort(stat_boots(:,1));
s2=sort(stat_boots(:,2));
s3=sort(stat_boots(:,3));
stats_99=[s1(round(0.99*size(s1,1))) s2(round(0.99*size(s2,1))) s3(round(0.99*size(s3,
;
stats_95=[s1(round(0.95*size(s1,1))) s2(round(0.95*size(s2,1))) s3(round(0.95*size(s3,
;
stats_90=[s1(round(0.90*size(s1,1))) s2(round(0.90*size(s2,1))) s3(round(0.90*size(s3,
if stat_0(1)>stats_95(1); num_ss=num_ss+1; end;
if stat_0(2)>stats_95(2); num_bp=num_bp+1; end;
if stat_0(3)>stats_95(3); num_cf=num_cf+1; end;
end
```