SSRMC Time Series Analysis

Topic One: Regression with Time Series Data (Stationary Variables)

STATA Codes and Outputs

1. <u>Finite Distributed Lags Model (Slides 12 – 14)</u>

use okun, clear
** tq() transforms the input into integer equivalent of the number of quarters have passed since the first one in 1960.*
** _n-1: increment the observations by 1*
generate date = tq(1985q2) + _n-1
list date in 1

	date
1.	101

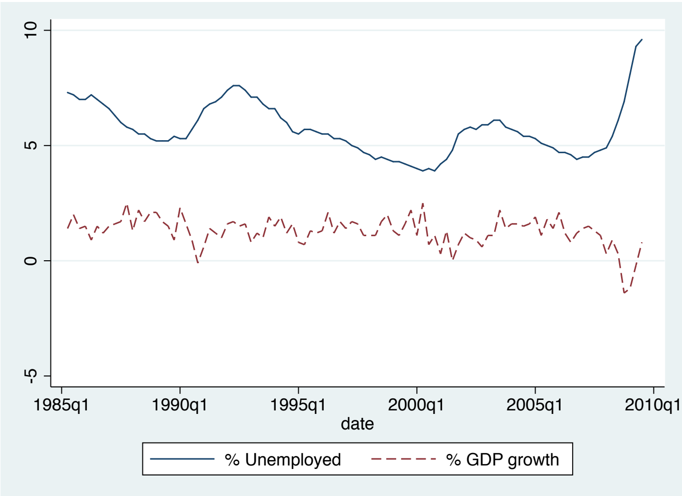
format %tq date
list date in 1

	date
1.	1985q2

* tsset declares the variable to be time-series.
tsset date

```
time variable:  date, 1985q2 to 2009q3
        delta:  1 quarter
```

label var u "% Unemployed"
label var g "% GDP growth"
tsline u g, lpattern(solid dash)



* L. : lag(-1)
* L2. : lag(-2)
* D. : first difference
* D. : difference of difference

list date u L.u D.u g L1.g L2.g L3.g in 1/5

	date	u	L. u	D. u	g	L. g	L2. g	L3. g
1.	1985q2	7.3	.	.	1.4	.	.	.
2.	1985q3	7.2	7.3	-.1	2	1.4	.	.
3.	1985q4	7	7.2	-.2	1.4	2	1.4	.
4.	1986q1	7	7	0	1.5	1.4	2	1.4
5.	1986q2	7.2	7	.2	.9	1.5	1.4	2

list date u L.u D.u g L1.g L2.g L3.g in 96/98

	date	u	L. u	D. u	g	L. g	L2. g	L3. g
96.	2009q1	8.1	6.9	1.2	-1.2	-1.4	.3	.9
97.	2009q2	9.3	8.1	1.2	-.2	-1.2	-1.4	.3
98.	2009q3	9.6	9.3	.3	.8	-.2	-1.2	-1.4

regress D.u L(0/3).g

Source	SS	df	MS			
				Number of obs	=	95
				F(4, 90)	=	42.23
Model	5.13367789	4	1.28341947	Prob > F	=	0.0000
Residual	2.73516422	90	.030390714	R-squared	=	0.6524
				Adj R-squared	=	0.6370
Total	7.86884211	94	.083711086	Root MSE	=	.17433

D.u	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
g						
--.	-.2020526	.0330131	-6.12	0.000	-.267639	-.1364663
L1.	-.1645352	.0358175	-4.59	0.000	-.2356929	-.0933774
L2.	-.071556	.0353043	-2.03	0.046	-.1416941	-.0014179
L3.	.003303	.0362603	0.09	0.928	-.0687345	.0753405
_cons	.5809746	.0538893	10.78	0.000	.4739142	.688035

regress D.u L(0/2).g

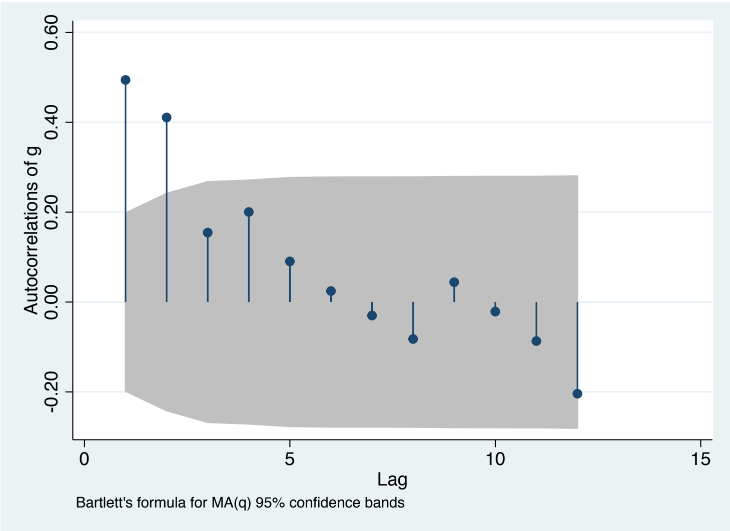
Source	SS	df	MS			
				Number of obs	=	96
				F(3, 92)	=	57.95
Model	5.17925206	3	1.72641735	Prob > F	=	0.0000
Residual	2.74074794	92	.029790739	R-squared	=	0.6539
				Adj R-squared	=	0.6427
Total	7.92	95	.083368421	Root MSE	=	.1726

D.u	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
g						
--.	-.2020216	.0323832	-6.24	0.000	-.2663374	-.1377059
L1.	-.1653269	.0335368	-4.93	0.000	-.2319339	-.0987198
L2.	-.0700135	.0331	-2.12	0.037	-.1357529	-.0042741
_cons	.5835561	.0472119	12.36	0.000	.4897892	.6773231

2. <u>The k-th order sample autocorrelation (Slides 16 - 18)</u>

```
* ac: computes sample autocorrelations
* lag(12): computes autocorrelations up to 12 periods apart
* generate(ac_g): save the autocorrelation coefficients in variable named ac_g

ac g, lags(12) generate(ac_g)
```



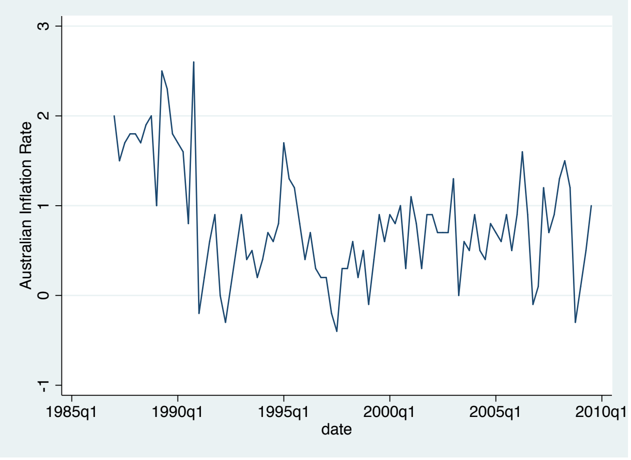

```
gen z=sqrt(e(N))*ac_g
list ac_g z in 1/12
```

	ac_g	z
1.	.49425676	4.842708
2.	.4107073	4.024093
3.	.1544205	1.513006
4.	.20043788	1.963882
5.	.09038538	.8855922
6.	.02447111	.239767
7.	-.03008434	-.2947652
8.	-.08231978	-.8065658
9.	.04410661	.4321548
10.	-.02128483	-.2085479
11.	-.08683463	-.8508022
12.	-.20404326	-1.999207

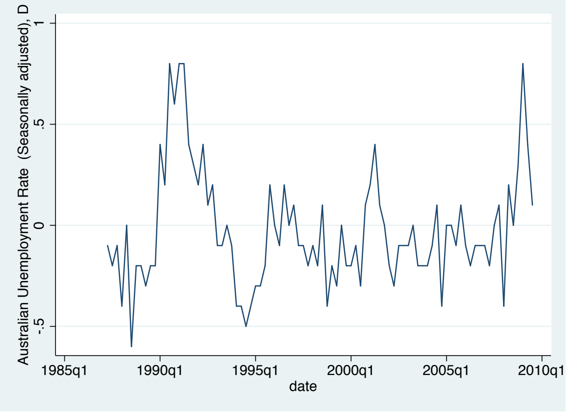
3. <u>Correlogram (Slides 18 - 21)</u>

```
use phillips_aus, clear
generate date = tq(1987q1) + _n-1
format %tq date
tsset date

tsline inf
```



tsline D.u



reg inf D.u

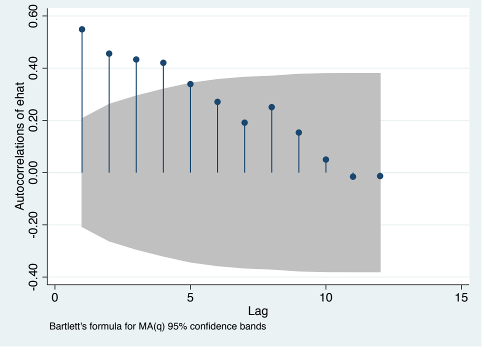
Source	SS	df	MS
Model	**2.04834633**	**1**	**2.04834633**
Residual	**34.0445426**	**88**	**.386869802**
Total	**36.0928889**	**89**	**.405538077**

Number of obs	=	**90**
F(1, 88)	=	**5.29**
Prob > F	=	**0.0238**
R-squared	=	**0.0568**
Adj R-squared	=	**0.0460**
Root MSE	=	**.62199**

inf	Coef.	Std. Err.	t	P>|t|	[95% Conf. Interval]	
u						
D1.	**-.5278638**	**.2294049**	**-2.30**	**0.024**	**-.9837578**	**-.0719699**
_cons	**.7776213**	**.0658249**	**11.81**	**0.000**	**.646808**	**.9084345**

* predict: save residuals in a variable named ehat
predict ehat, res

ac ehat, lags(12) generate(rk)



list rk in 1/5

```
        rk
 1.  .54865864
 2.  .45573248
 3.  .43321579
 4.  .42049358
 5.  .33903419
```

4. <u>LM tests (Slides 22 - 24)</u>

reg inf D.u
predict ehat, res

* LM test for AR(1) : Method 1 (delete the first observation)
quietly regress ehat D.u L.ehat
di "Observations = " e(N) " and TR2 = " e(N)*e(r2)

```
. di "Observations = " e(N) " and TR2 = " e(N)*e(r2)
Observations = 89 and TR2 = 27.608808
```

* LM test for AR(1) : Method 2 (Replace ehat[1] with zero)
replace ehat = 0 in 1
quietly regress ehat D.u L.ehat
di "Observations = " e(N) " and TR2 = " e(N)*e(r2)
drop ehat

```
. di "Observations = " e(N) " and TR2 = " e(N)*e(r2)
Observations = 90 and TR2 = 27.592347
```

* LM test for AR(4): Method 1 (delete the first four observations)
reg inf D.u
predict ehat, res

quietly regress ehat D.u L(1/4).ehat
di "Observations = " e(N) " and TR2 = " e(N)*e(r2)

```
. di "Observations = " e(N) " and TR2 = " e(N)*e(r2)
Observations = 86 and TR2 = 33.385269
```

* LM test for AR(4): Method 2 (Replace ehat[-3] to ehat[1] with zero such that the first four lagged terms can be used in the regression)
set obs 94 // add 3 observations to data
gsort -date // moves missing observations to end
replace date = date[_n-1] - 1 if missing(date) // creates dates for missing obs
replace ehat = 0 if missing(ehat) // puts zeros in for missing ehats
sort date // re-sort data into ascending order

regress ehat D.u L(1/4).ehat
di "Observations = " e(N) " and TR2 = " e(N)*e(r2)

```
. di "Observations = " e(N) " and TR2 = " e(N)*e(r2)
Observations = 90 and TR2 = 36.671898
```

* Using the built-in bgodfrey command to test the AR(1) and AR(4) alternatives
regress inf D.u
predict ehat, res
estat bgodfrey, lags(1)

Breusch-Godfrey LM test for autocorrelation

lags(p)	chi2	df	Prob > chi2
1	27.592	1	0.0000

H0: no serial correlation

estat bgodfrey, lags(4)

Breusch-Godfrey LM test for autocorrelation

lags(p)	chi2	df	Prob > chi2
4	36.672	4	0.0000

H0: no serial correlation

5. <u>OLS with HAC standard errors (Slides 27 - 28)</u>

* calculate bandwidth (the number of lags in the next step)
scalar B = round(4*(e(N)/100)^(2/9))
scalar list B

* calculate and store the standard OLS coefficient estimates
regress inf D.u
estimates store Wrong_SE //estimate names are case sensitive

Source	SS	df	MS
Model	2.04834633	1	2.04834633
Residual	34.0445426	88	.386869802
Total	36.0928889	89	.405538077

Number of obs	=	90
F(1, 88)	=	5.29
Prob > F	=	0.0238
R-squared	=	0.0568
Adj R-squared	=	0.0460
Root MSE	=	.62199

inf	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
u						
D1.	-.5278638	.2294049	-2.30	0.024	-.9837578	-.0719699
_cons	.7776213	.0658249	11.81	0.000	.646808	.9084345

* calculate and store the HAC coefficient estimates
newey inf D.u, lag(4)
estimates store HAC_4

```
Regression with Newey-West standard errors          Number of obs     =         90
maximum lag: 4                                      F(  1,        88) =       2.76
                                                    Prob > F          =     0.1001

------------------------------------------------------------------------------
             |             Newey-West
         inf |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
           u |
         D1. |  -.5278638   .3176735    -1.66   0.100    -1.159173    .1034454
             |
       _cons |   .7776213   .1116107     6.97   0.000     .5558184    .9994242
------------------------------------------------------------------------------
```

* Display results
ssc install estout, replace // install estout to use the following command
esttab Wrong_SE HAC_4, title("Dependent Variable: inf") mtitles("LS" "HAC(4)") scalars (r2 r2_a rss aic), using output.rtf, append

```
Dependent Variable: inf
--------------------------------------------
                      (1)             (2)
                       LS          HAC(4)
--------------------------------------------
D.u                -0.528*         -0.528
                  (-2.30)         (-1.66)

_cons               0.778***        0.778***
                  (11.81)          (6.97)
--------------------------------------------
N                      90              90
r2                 0.0568
r2_a               0.0460
rss                 34.04
aic                 171.9               .
--------------------------------------------
t statistics in parentheses
* p<0.05, ** p<0.01, *** p<0.001
```

6. <u>Nonlinear least squares of AR(1) regression model (Slide 31)</u>

* nl: nonlinear regression
* You must enclose the entire equation in parentheses, each parameter in braces, and all variables in the *variables(varlist)* part.

nl (inf = {b1}*(1-{rho}) + {b2}*D.u + {rho}*L.inf - {rho}*{b2}*L.D.u), variables(inf D.u L.inf L.D.u)
estimates store NL

```
Iteration 0:  residual SS =  26.75696
Iteration 1:  residual SS =  23.21352
Iteration 2:  residual SS =  23.19868
Iteration 3:  residual SS =  23.19868
Iteration 4:  residual SS =  23.19868
Iteration 5:  residual SS =  23.19868
```

```
      Source |       SS           df       MS
-------------+----------------------------------    Number of obs =        89
       Model |  12.386043          2  6.19302165    R-squared     =    0.3481
    Residual |  23.198676         86  .269752044    Adj R-squared =    0.3329
-------------+----------------------------------    Root MSE      =  .5193766
       Total |  35.584719         88  .404371808    Res. dev.     =  132.9069

------------------------------------------------------------------------------
         inf |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         /b1 |   .7608716   .1245311     6.11   0.000      .513312    1.008431
        /rho |   .5573922   .0901546     6.18   0.000     .3781709    .7366136
         /b2 |   -.694388    .247894    -2.80   0.006    -1.187185    -.201591
------------------------------------------------------------------------------
  Parameter b1 taken as constant term in model & ANOVA table
```

7. ARDL(1,1) and ARDL(1,0) models (Slides 32 – 34)

regress inf L.inf D.u L.D.u
estimates store ARDL_1_1

```
      Source |       SS           df       MS      Number of obs   =        89
-------------+----------------------------------   F(3, 85)        =     15.18
       Model |  12.4166337         3  4.13887791   Prob > F        =    0.0000
    Residual |  23.1680854        85   .27256571   R-squared       =    0.3489
-------------+----------------------------------   Adj R-squared   =    0.3260
       Total |  35.5847191        88  .404371808   Root MSE        =    .52208

------------------------------------------------------------------------------
         inf |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         inf |
         L1. |   .5592676   .0907962     6.16   0.000     .3787403    .7397948
             |
           u |
         D1. |  -.6881852   .2498704    -2.75   0.007    -1.184994    -.191376
         LD. |   .3199526    .257504     1.24   0.217    -.1920343    .8319396
             |
       _cons |   .3336325   .0899028     3.71   0.000     .1548817    .5123834
------------------------------------------------------------------------------
```

testnl _b[L.D.u]=-_b[L.inf]*_b[D.u]

```
      chi2(1) =        0.11
  Prob > chi2 =        0.7376
```

regress inf L.inf D.u
estimates store ARDL_1_0

Source	SS	df	MS			
				Number of obs	=	90
				F(2, 87)	=	23.05
Model	12.5023522	2	6.25117612	Prob > F	=	0.0000
Residual	23.5905366	87	.271155594	R-squared	=	0.3464
				Adj R-squared	=	0.3314
Total	36.0928889	89	.405538077	Root MSE	=	.52073

inf	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
inf						
L1.	.5282472	.0850756	6.21	0.000	.3591502	.6973443
u						
D1.	-.4908647	.1921491	-2.55	0.012	-.872782	-.1089475
_cons	.3547951	.0876023	4.05	0.000	.180676	.5289142

esttab Wrong_SE HAC_4 NL ARDL_1_1 ARDL_1_0, scalars(r2_a rss aic) mtitles("OLS" "HAC(4)" "Nonlinear" "ARDL (1,1)" " ARDL (0,1)"), using outputs.rft, replace

	(1)	(2)	(3)	(4)	(5)
	OLS	HAC(4)	Nonlinear	ARDL (1,1)	ARDL (0,1)
main					
D.u	-0.528*	-0.528		-0.688**	-0.491*
	(-2.30)	(-1.66)		(-2.75)	(-2.55)
LD.u				0.320	
				(1.24)	
L.inf				0.559***	0.528***
				(6.16)	(6.21)
_cons	0.778***	0.778***	0.761***	0.334***	0.355***
	(11.81)	(6.97)	(6.11)	(3.71)	(4.05)
rho					
_cons			0.557***		
			(6.18)		
b2					
_cons			-0.694**		
			(-2.80)		
N	90	90	89	89	90
r2_a	0.0460		0.333	0.326	0.331
rss	34.04		23.20	23.17	23.59
aic	171.9	.	138.9	140.8	140.9

t statistics in parentheses
* p<0.05, ** p<0.01, *** p<0.001

8. <u>Autoregressive Distributed Lag (ARDL) Models (Slide 37 – 40)</u>

* Philips Curve Example:

```
quietly regress inf L.inf D.u
estat bgodfrey, lags(1 2 3 4 5)
```

Breusch-Godfrey LM test for autocorrelation

lags(p)	chi2	df	Prob > chi2
1	4.130	1	0.0421
2	5.123	2	0.0772
3	5.221	3	0.1563
4	9.554	4	0.0486
5	12.485	5	0.0287

H0: no serial correlation

* forvalues: loop over consecutive values in commands included in braces
* quietly: suppress outputs
* scalar: defines the contents of a scalar variable (numerial or string value)
* if date >= tq(1988q3): make sure sample size is the same in each step.

```
forvalues q=0/1 {
  forvalues p=1/6 {
    quietly regress inf L(1/`p').inf L(0/`q').D.u if date >= tq(1988q3)
    display "p=`p'  q=`q'"
    scalar aic = ln(e(rss)/e(N))+2*e(rank)/e(N)
    scalar sc = ln(e(rss)/e(N))+e(rank)*ln(e(N))/e(N)
    scalar obs = e(N)
    scalar list aic sc obs
    }
  }
```

```
p=1  q=0
       aic = -1.2466292
        sc =  -1.160418
       obs =         85
p=2  q=0
       aic = -1.2904903
        sc =  -1.175542
       obs =         85
p=3  q=0
       aic = -1.3352266
        sc = -1.1915413
       obs =         85
p=4  q=0
       aic = -1.4019823
        sc = -1.2295599
       obs =         85
p=5  q=0
       aic = -1.3963808
        sc = -1.1952213
       obs =         85
p=6  q=0
       aic = -1.3778837
        sc = -1.1479872
       obs =         85
```

```
p=1  q=1
       aic = -1.2424601
        sc = -1.1275118
       obs =         85
p=2  q=1
       aic = -1.2860299
        sc = -1.1423446
       obs =         85
p=3  q=1
       aic = -1.3233286
        sc = -1.1509061
       obs =         85
p=4  q=1
       aic = -1.3795327
        sc = -1.1783732
       obs =         85
p=5  q=1
       aic = -1.3729049
        sc = -1.1430084
       obs =         85
p=6  q=1
       aic =  -1.354396
        sc = -1.0957623
       obs =         85
```

regress L(0/4).inf D.u

```
      Source |       SS           df       MS      Number of obs   =        87
-------------+----------------------------------   F(5, 81)        =     13.71
       Model |  15.4337676         5  3.08675353   Prob > F        =    0.0000
    Residual |  18.2333588        81  .225103195   R-squared       =    0.4584
-------------+----------------------------------   Adj R-squared   =    0.4250
       Total |  33.6671264        86  .391478214   Root MSE        =    .47445

------------------------------------------------------------------------------
         inf |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         inf |
         L1. |   .2354401   .1015555     2.32   0.023     .0333765    .4375037
         L2. |    .121328   .1037571     1.17   0.246    -.0851161    .3277722
         L3. |   .1676895   .1049597     1.60   0.114    -.0411473    .3765264
         L4. |   .2819156   .1013801     2.78   0.007      .080201    .4836302
             |
           u |
         D1. |  -.7901718   .1885334    -4.19   0.000    -1.165294   -.4150496
             |
       _cons |   .1000999   .0982599     1.02   0.311    -.0954064    .2956063
------------------------------------------------------------------------------
```

```
Breusch-Godfrey LM test for autocorrelation
---------------------------------------------------------------------------
    lags(p)  |          chi2               df                 Prob > chi2
-------------+-------------------------------------------------------------
       1     |          0.504               1                   0.4777
       2     |          2.566               2                   0.2773
       3     |          2.797               3                   0.4240
       4     |          6.721               4                   0.1514
       5     |          6.793               5                   0.2365
---------------------------------------------------------------------------
                        H0: no serial correlation
```

* Okun's Law Example:

use okun, clear
generate date = tq(1985q2) + _n-1
format %tq date
tsset date

reg D.u L(0/2).g

```
      Source |       SS           df       MS      Number of obs   =        96
-------------+----------------------------------   F(3, 92)        =     57.95
       Model |  5.17925206         3  1.72641735   Prob > F        =    0.0000
    Residual |  2.74074794        92  .029790739   R-squared       =    0.6539
-------------+----------------------------------   Adj R-squared   =    0.6427
       Total |        7.92        95  .083368421   Root MSE        =     .1726

------------------------------------------------------------------------------
         D.u |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
           g |
         --. |  -.2020216   .0323832    -6.24   0.000    -.2663374   -.1377059
         L1. |  -.1653269   .0335368    -4.93   0.000    -.2319339   -.0987198
         L2. |  -.0700135      .0331    -2.12   0.037    -.1357529   -.0042741
             |
       _cons |   .5835561   .0472119    12.36   0.000     .4897892    .6773231
------------------------------------------------------------------------------
```

estat bgodfrey, lags(1 2 3 4 5)

Breusch-Godfrey LM test for autocorrelation

lags(p)	chi2	df	Prob > chi2
1	12.364	1	0.0004
2	12.894	2	0.0016
3	13.754	3	0.0033
4	15.228	4	0.0043
5	19.648	5	0.0015

H0: no serial correlation

```
forvalues q=1/3 {
  forvalues p=0/2 {
    quietly regress L(0/`p').D.u L(0/`q').g if date >= tq(1986q1)
    display "p=`p'  q=`q'"
    scalar aic = ln(e(rss)/e(N))+2*e(rank)/e(N)
    scalar sc = ln(e(rss)/e(N))+e(rank)*ln(e(N))/e(N)
    scalar obs = e(N)
    scalar list aic sc obs
    }
  }
```

```
p=0  q=1
       aic = -3.4362364
        sc = -3.3555876
       obs =         95
p=1  q=1
       aic = -3.5879866
        sc =  -3.480455
       obs =         95
p=2  q=1
       aic = -3.5693074
        sc = -3.4348928
       obs =         95
p=0  q=2
       aic = -3.4633827
        sc =  -3.355851
       obs =         95
p=1  q=2
       aic = -3.5675498
        sc = -3.4331352
       obs =         95
p=2  q=2
       aic = -3.5483196
        sc = -3.3870221
       obs =         95
p=0  q=3
       aic = -3.4424223
        sc = -3.3080077
       obs =         95
p=1  q=3
       aic = -3.5611594
        sc = -3.3998619
       obs =         95
p=2  q=3
       aic = -3.5490965
        sc = -3.3609161
       obs =         95
```

reg D.u L.D.u L(0/1).g

Source	SS	df	MS
Model	5.49727601	3	1.83242534
Residual	2.42272399	92	.026333956
Total	7.92	95	.083368421

Number of obs	=	96
F(3, 92)	=	69.58
Prob > F	=	0.0000
R-squared	=	0.6941
Adj R-squared	=	0.6841
Root MSE	=	.16228

D.u	Coef.	Std. Err.	t	P>|t|	[95% Conf. Interval]	
u						
LD.	.3501158	.084573	4.14	0.000	.1821466	.518085
g						
--.	-.1840843	.0306984	-6.00	0.000	-.245054	-.1231146
L1.	-.0991552	.0368244	-2.69	0.008	-.1722917	-.0260187
_cons	.3780104	.0578398	6.54	0.000	.2631356	.4928853

estat bgodfrey, lags(1 2 3 4 5)

Breusch-Godfrey LM test for autocorrelation

lags(*p*)	chi2	df	Prob > chi2
1	0.170	1	0.6804
2	0.271	2	0.8731
3	3.896	3	0.2729
4	6.141	4	0.1889
5	8.226	5	0.1442

H0: no serial correlation

* When ACI and SC do not agree:

```
forvalues p=1/5 {
  qui reg g L(1/`p').g if date> tq(1986q2)
  display "p=`p'"
      scalar aic = ln(e(rss)/e(N))+2*e(rank)/e(N)
      scalar sc = ln(e(rss)/e(N))+e(rank)*ln(e(N))/e(N)
      scalar obs = e(N)
      scalar list aic sc obs
 }
```

```
p=1
       aic = -1.0935183
        sc = -1.0390538
       obs =         93
p=2
       aic =  -1.130582
        sc = -1.0488852
       obs =         93
p=3
       aic = -1.1242025
        sc = -1.0152735
       obs =         93
p=4
       aic = -1.1331587
        sc = -.99699743
       obs =         93
p=5
       aic = -1.1116622
        sc = -.94826871
       obs =         93
```

reg g L(1/2).g

Source	SS	df	MS
Model	11.6417916	2	5.82089582
Residual	28.4081042	93	.305463486
Total	40.0498958	95	.421577851

Number of obs	=	96
F(2, 93)	=	19.06
Prob > F	=	0.0000
R-squared	=	0.2907
Adj R-squared	=	0.2754
Root MSE	=	.55269

g	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
g						
L1.	.3770015	.100021	3.77	0.000	.1783797	.5756233
L2.	.2462394	.1028688	2.39	0.019	.0419623	.4505165
_cons	.4657262	.1432576	3.25	0.002	.181245	.7502073

estat bgodfrey, lags(1 2 3 4 5)

Breusch-Godfrey LM test for autocorrelation

lags(*p*)	chi2	df	Prob > chi2
1	1.663	1	0.1972
2	2.653	2	0.2654
3	3.799	3	0.2840
4	4.904	4	0.2973
5	5.828	5	0.3233

H0: no serial correlation

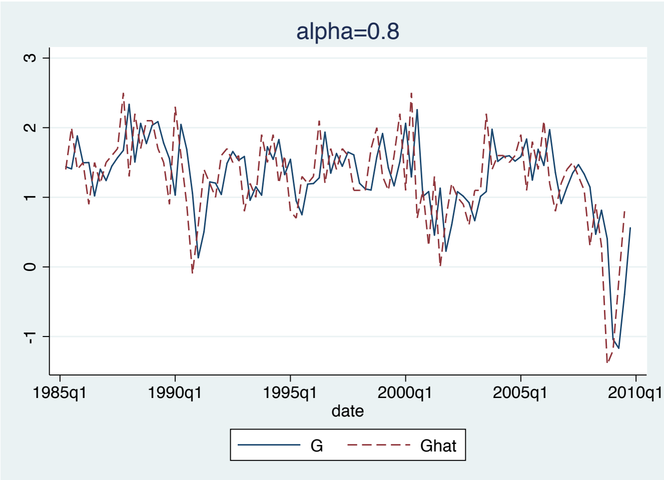
9. Exponential Smoothing (Slides 44 – 46)

use okun, clear
generate date = tq(1985q2) + _n-1
format %tq date
tsset date

tsappend, add(1)
tssmooth exponential sm2=g, parms(.8)

```
exponential coefficient  =      0.8000
sum-of-squared residuals =      35.452
root mean squared error  =      .60146
```

tsline sm2 g, legend(lab (1 "G") lab(2 "Ghat")) title(alpha=0.8) lpattern(solid dash)



scalar f2 = .8*g[98]+(1-.8)*sm2[98]
scalar list f2

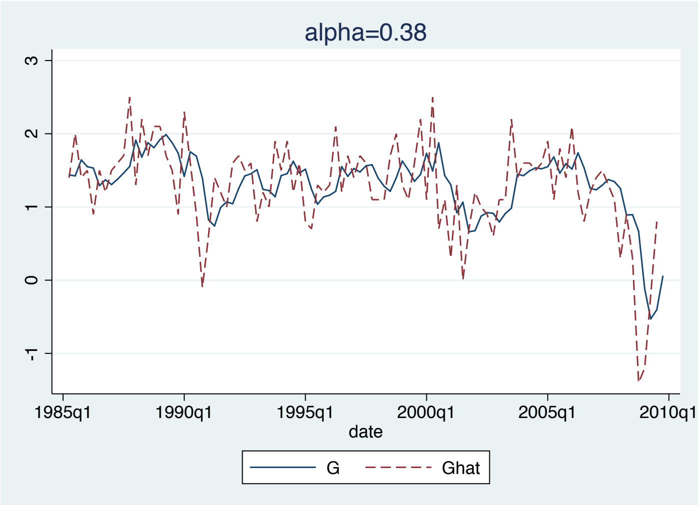
```
 f2 =  .56128444
```

tssmooth exponential sm3=g

```
computing optimal exponential  coefficient (0,1)

optimal exponential coefficient =        0.3803
sum-of-squared residuals        =     31.122043
root mean squared error         =     .56353515
```

tsline sm3 g, legend(lab (1 "G") lab(2 "Ghat")) title(alpha=0.38) lpattern(solid dash)



scalar f3 = r(alpha)*g[98]+(1-r(alpha))*sm3[98]
scalar list f3

```
f3 =  .05367152
```

list sm3 in 99

	sm3
99.	.0536715

10. <u>Impact and Delay Multipliers from Okun's ARDL(1,1) model (Slides 49 & 50)</u>

```
regress D.u L.D.u L(0/1).g

scalar b0 = _b[g]
scalar b1 = _b[L1.D.u]*b0+_b[L1.g]
scalar b2 = b1*_b[L1.D.u]
scalar list b0 b1 b2
 b0 = -.18408429
 b1 = -.16360601
 b2 = -.05728104
```

```
* An alternative method: Exploiting variable creation
regress D.u L.D.u L(0/1).g
gen mult = _b[g] in 1
replace mult = L.mult*_b[L1.D.u]+_b[L1.g] in 2
replace mult = L.mult*_b[L1.D.u] in 3/8
list mult in 1/8
```

	mult
1.	-.1840843
2.	-.163606
3.	-.057281
4.	-.020055
5.	-.0070216
6.	-.0024584
7.	-.0008607
8.	-.0003013

```
gen lag = _n-1 in 1/8
line mult lag in 1/8
```