```
  ___  ____  ____  ____  ____ (R)
 /__    /   ____/   /   ____/
___/   /   /___/   /   /___/   16.1   Copyright 1985-2019 StataCorp LLC
  Statistics/Data Analysis            StataCorp
                                      4905 Lakeway Drive
     Special Edition                  College Station, Texas 77845 USA
                                      800-STATA-PC        http://www.stata.com
                                      979-696-4600        stata@stata.com
                                      979-696-4601 (fax)

Unlimited-user Stata network license expires 23 Oct 2020:
       Serial number:  401609000311
         Licensed to:  Arjun Jain
                       UW-Madison

Notes:
      1.  Unicode is supported; see help unicode_advice.
      2.  Maximum number of variables is set to 5000; see help set_maxvar.
      3.  New update available; type -update all-

 1 . do "C:\z_toshiba\course work\phd\econ 761\hw\hw2\hw2.do"

 2 . // code for questions 2 and 3 of hw2
 3 .
 4 . // clear workspace
 5 . clear

 6 .
 7 . // 2a (setup)
 8 .
 9 . // for each city, number of firms uniformly distributed on {1,2,...,10}
10 . set obs 1000
   number of observations (_N) was 0, now 1,000

11 . gen unif = runiform()

12 . gen num_firms = int(unif*10+1)

13 .
14 . // in 500 cities, firms collude perfectly when num_firms <= 8
15 . gen collude = 0

16 . replace collude = 1 if _n <= 500 & num_firms <= 8
   (400 real changes made)

17 .
18 . // 2b (construct L_i, H, e)
19 .
20 . // demand function parameter initialization
21 . gen c0 = 1

22 . gen c1 = 0.9
```

```
. gen xi = 0

.
. // cost function parameter initialization
. gen F = 1

. gen b0 = 1

. gen b1 = 0

. gen eta = 0

.
. // construct Lerner index, Herfindahl index, demand elasticity
. gen lerner_cournot = c1/num_firms

. gen lerner_monopoly = c1

. gen lerner = collude*lerner_monopoly + (1-collude)*lerner_cournot

. gen observed_lerner = ln(lerner) + 0.1*(unif - 0.5)

. gen herfindahl = 1/num_firms

. gen elasticity = 1/c1

.
. // 2c (regressions and tests)
.
. // structure-conduct-performance paradigm regressions
. gen ln_herfindahl = ln(herfindahl)

. regress observed_lerner ln_herfindahl if _n <= 500  // collusion is possible

      Source |       SS           df       MS      Number of obs   =       500
-------------+----------------------------------   F(1, 498)       =    163.08
       Model |  95.6248438         1  95.6248438   Prob > F        =    0.0000
    Residual |  292.006505       498  .586358443   R-squared       =    0.2467
-------------+----------------------------------   Adj R-squared   =    0.2452
       Total |  387.631349       499   .77681633   Root MSE        =    .76574

------------------------------------------------------------------------------
observed_le~r |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
--------------+----------------------------------------------------------------
ln_herfindahl |   .6193705   .0485006    12.77   0.000     .5240795    .7146614
        _cons |   .3871213   .0812891     4.76   0.000     .2274094    .5468331
------------------------------------------------------------------------------

. test ln_herfindahl = 1

 ( 1)  ln_herfindahl = 1

       F(  1,   498) =   61.59
            Prob > F =    0.0000
```

```
45 . regress observed_lerner ln_herfindahl if _n > 500  // no collusion

      Source |       SS           df       MS      Number of obs   =       500
-------------+----------------------------------   F(1, 498)       >  99999.00
       Model |  236.102625         1  236.102625   Prob > F        =    0.0000
    Residual |  .043826911       498  .000088006   R-squared       =    0.9998
-------------+----------------------------------   Adj R-squared   =    0.9998
       Total |  236.146452       499  .473239383   Root MSE        =    .00938

------------------------------------------------------------------------------
observed_le~r |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
ln_herfindahl |   .9616705   .0005871  1637.93   0.000      .960517    .9628241
        _cons |  -.1633087   .0009573  -170.59   0.000    -.1651896   -.1614278
------------------------------------------------------------------------------

46 . test ln_herfindahl = 1

    ( 1)  ln_herfindahl = 1

          F(  1,   498) = 4261.89
               Prob > F =    0.0000

47 . regress observed_lerner ln_herfindahl  // pooled sample

      Source |       SS           df       MS      Number of obs   =     1,000
-------------+----------------------------------   F(1, 998)       =    500.45
       Model |  294.916379         1  294.916379   Prob > F        =    0.0000
    Residual |   588.12952       998  .589308137   R-squared       =    0.3340
-------------+----------------------------------   Adj R-squared   =    0.3333
       Total |  883.045899       999  .883929829   Root MSE        =    .76766

------------------------------------------------------------------------------
observed_le~r |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
ln_herfindahl |   .7639598   .0341501    22.37   0.000     .6969455    .8309741
        _cons |   .0769168    .056465     1.36   0.173    -.0338869    .1877205
------------------------------------------------------------------------------

48 . test ln_herfindahl = 1

    ( 1)  ln_herfindahl = 1

          F(  1,   998) =   47.77
               Prob > F =    0.0000

49 .
50 . // 2d (repeat 2b and 2c for linear demand)
51 .
52 . // demand function parameter initialization
53 . gen a0 = 3

54 . gen a1 = 1
```

```
55 . gen nu = 0

56 .
57 . // construct Lerner index, Herfindahl index, demand elasticity
58 . gen lerner_cournot2 = (a0+nu-b0-eta)/(a0+nu+num_firms*(b0+eta))

59 . gen lerner_monopoly2 = (a0+nu-b0-eta)/(a0+nu+b0+eta)

60 . gen lerner2 = collude*lerner_monopoly2 + (1-collude)*lerner_cournot2

61 . gen observed_lerner2 = ln(lerner2) + 0.1*(unif - 0.5)

62 . gen herfindahl2 = 1/num_firms

63 . gen elasticity2 = (a0+nu+b0+eta)/(a0+nu-b0-eta)

64 .
65 . // structure-conduct-performance paradigm regressions
66 . gen ln_herfindahl2 = ln(herfindahl2)

67 . regress observed_lerner2 ln_herfindahl2 if _n <= 500  // collusion is possible

      Source |       SS           df       MS      Number of obs   =       500
-------------+----------------------------------   F(1, 498)       =    146.01
       Model |  21.5861585         1  21.5861585   Prob > F        =    0.0000
    Residual |  73.6262089       498  .147843793   R-squared       =    0.2267
-------------+----------------------------------   Adj R-squared   =    0.2252
       Total |  95.2123673       499  .190806347   Root MSE        =     .3845

------------------------------------------------------------------------------
observed_ler~2 |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
---------------+--------------------------------------------------------------
ln_herfindahl2 |   .2942746   .0243538    12.08   0.000     .2464258    .3421235
         _cons |  -.4726272    .040818   -11.58   0.000     -.552824   -.3924304
------------------------------------------------------------------------------

68 . test ln_herfindahl2 = 1

 ( 1)  ln_herfindahl2 = 1

       F(  1,   498) =  839.73
            Prob > F =    0.0000

69 . regress observed_lerner2 ln_herfindahl2 if _n > 500  // no collusion

      Source |       SS           df       MS      Number of obs   =       500
-------------+----------------------------------   F(1, 498)       =  24124.38
       Model |  58.8244536         1  58.8244536   Prob > F        =    0.0000
    Residual |  1.21431447       498  .002438382   R-squared       =    0.9798
-------------+----------------------------------   Adj R-squared   =    0.9797
       Total |  60.0387681       499  .120318173   Root MSE        =    .04938

------------------------------------------------------------------------------
observed_ler~2 |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
---------------+--------------------------------------------------------------
ln_herfindahl2 |   .4800151   .0030905   155.32   0.000     .4739431    .4860871
         _cons |  -.6577959   .0050391  -130.54   0.000    -.6676963   -.6478954
------------------------------------------------------------------------------
```

```
70 . test ln_herfindahl2 = 1

 ( 1)  ln_herfindahl2 = 1

       F(  1,   498) =28309.22
            Prob > F =    0.0000

71 . regress observed_lerner2 ln_herfindahl2  // pooled sample

      Source |       SS           df       MS      Number of obs   =     1,000
-------------+----------------------------------   F(1, 998)       =    537.11
       Model |  71.3593422         1  71.3593422   Prob > F        =    0.0000
    Residual |  132.591543       998  .132857258   R-squared       =    0.3499
-------------+----------------------------------   Adj R-squared   =    0.3492
       Total |  203.950885       999   .20415504   Root MSE        =     .3645

--------------------------------------------------------------------------------
observed_ler~2 |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
---------------+----------------------------------------------------------------
ln_herfindahl2 |   .3757911   .0162149    23.18   0.000     .3439719    .4076103
         _cons |  -.5796315   .0268102   -21.62   0.000    -.6322424   -.5270206
--------------------------------------------------------------------------------

72 . test ln_herfindahl2 = 1

 ( 1)  ln_herfindahl2 = 1

       F(  1,   998) = 1481.95
            Prob > F =    0.0000

73 .
74 . // 3a (setup, regressions, construct num_firms, L_i, H)
75 .
76 . // clear workspace
77 . clear

78 .
79 . // 1000 cities
80 . set obs 1000
number of observations (_N) was 0, now 1,000

81 . gen unif = runiform()

82 .
83 . // demand function parameter initialization
84 . gen a0 = 5

85 . gen a1 = 1

86 . gen nu = 2*(unif - 0.5)

87 .
88 . // cost function parameter initialization
```

```
. gen F = 1

. gen b0 = 1

. gen b1 = 0

. gen eta = 0

. 
. // firms enter until profits are zero
. gen num_firms = (a0+nu-b0-eta-sqrt(F*a1))/(sqrt(F*a1))

. 
. // construct Lerner index, Herfindahl index, demand elasticity
. gen lerner = (a0+nu-b0-eta)/(a0+nu+num_firms*(b0+eta))

. gen observed_lerner = ln(lerner) + 0.1*(unif - 0.5)

. gen herfindahl = 1/num_firms

. gen elasticity = (a0+nu+b0+eta)/(a0+nu-b0-eta)

. 
. // structure-conduct-performance paradigm regression
. gen ln_herfindahl = ln(herfindahl)

. regress observed_lerner ln_herfindahl

      Source |       SS           df       MS      Number of obs   =     1,000
-------------+----------------------------------   F(1, 998)       >  99999.00
       Model |  .81922049         1  .81922049    Prob > F        =    0.0000
    Residual |  .006599608       998  6.6128e-06   R-squared       =    0.9920
-------------+----------------------------------   Adj R-squared   =    0.9920
       Total |  .825820098       999  .000826647   Root MSE        =    .00257

------------------------------------------------------------------------------
observed_le~r |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
--------------+----------------------------------------------------------------
ln_herfindahl |  -.1458873   .0004145  -351.97   0.000    -.1467007    -.145074
        _cons |  -.8506153   .0004606 -1846.81   0.000    -.8515191   -.8497114
------------------------------------------------------------------------------

. 
. // 3b (repeat 3a for new eta and nu)
. 
. // parameter initialization
. drop nu

. drop eta

. gen nu = 0

. gen eta = 2*(unif - 0.5)
```

```
114 .
115 . // firms enter until profits are zero
116 . gen num_firms2 = (a0+nu-b0-eta-sqrt(F*a1))/(sqrt(F*a1))

117 .
118 . // construct Lerner index, Herfindahl index, demand elasticity
119 . gen lerner2 = (a0+nu-b0-eta)/(a0+nu+num_firms2*(b0+eta))

120 . gen observed_lerner2 = ln(lerner2) + 0.1*(unif - 0.5)

121 . gen herfindahl2 = 1/num_firms2

122 . gen elasticity2 = (a0+nu+b0+eta)/(a0+nu-b0-eta)

123 .
124 . // structure-conduct-performance paradigm regression
125 . gen ln_herfindahl2 = ln(herfindahl2)

126 . regress observed_lerner2 ln_herfindahl2

      Source |       SS           df       MS      Number of obs   =     1,000
-------------+----------------------------------   F(1, 998)       =  16414.98
       Model |  72.1224304         1  72.1224304   Prob > F        =    0.0000
    Residual |  4.38490729       998  .004393695   R-squared       =    0.9427
-------------+----------------------------------   Adj R-squared   =    0.9426
       Total |  76.5073377       999  .076583922   Root MSE        =    .06628

------------------------------------------------------------------------------
observed_ler~2 |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
---------------+----------------------------------------------------------------
ln_herfindahl2 |  -1.359395   .0106102  -128.12   0.000    -1.380216   -1.338575
         _cons |  -2.116006   .0114953  -184.08   0.000    -2.138564   -2.093448
--------------------------------------------------------------------------------

127 .
    end of do-file

128 .
```