

Asymptotic Properties

Exogeneity Condition – Endogeneity Bias

Monte Carlo Simulation

```
. set obs 100
obs was 0, now 100

. set seed 54321

. g x=rnormal(0,1)

. g true_y=1+2*x

. set seed 98765

. g e=rnormal(0,1)

. g e2=0.01*x+e

. g e3=0.1*x+e

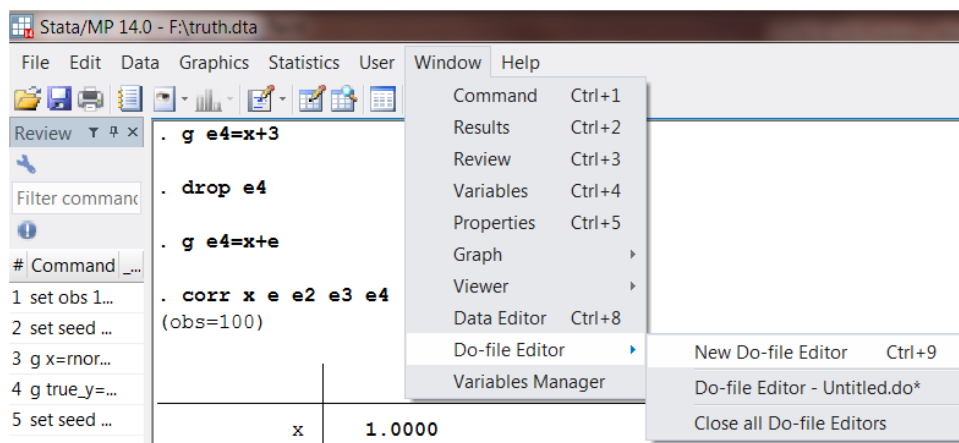
. g e4=x+e

. corr x e e2 e3 e4
(obs=100)
```

	x	e	e2	e3	e4
x	1.0000				
e	0.0796	1.0000			
e2	0.0896	1.0000	1.0000		
e3	0.1776	0.9951	0.9961	1.0000	
e4	0.7355	0.7340	0.7407	0.7974	1.0000

```
. save truth
file truth.dta saved
```

To create Do-file, go to Window, choose Do-file Editor, then, select New Do-file Ctrl+8.



Then, type in the command program in Do-file Editor. After finish, go to Tools, then, choose Run.

```
program MC01
  use truth, clear
  gen y = true_y + rnormal(0,1)
```



```

Simulations (10)
-----+----- 1 -----+----- 2 -----+----- 3 -----+----- 4 -----+----- 5
.....
. mean _b_x _b_cons
Mean estimation                Number of obs    =      10
-----+-----
              |          Mean   Std. Err.   [95% Conf. Interval]
-----+-----
      _b_x |    2.057198   .0457031    1.95381    2.160586
      _b_cons |    .993492   .0322138    .9206194    1.066365
-----+-----

```

```
. run "L:\Econometrics\Manual STATA\Do-Files\MC03.do"
```

```
. simulate _b _se, reps(10): MC03
```

```
command: MC03
```

```

Simulations (10)
-----+----- 1 -----+----- 2 -----+----- 3 -----+----- 4 -----+----- 5
.....
. mean _b_x _b_cons
Mean estimation                Number of obs    =      10
-----+-----
              |          Mean   Std. Err.   [95% Conf. Interval]
-----+-----
      _b_x |    2.121299   .0245532    2.065756    2.176842
      _b_cons |    1.032642   .0303892    .9638965    1.101387
-----+-----

```

```
. run "L:\Econometrics\Manual STATA\Do-Files\MC04.do"
```

```
. simulate _b _se, reps(10): MC04
```

```
command: MC04
```

```

Simulations (10)
-----+----- 1 -----+----- 2 -----+----- 3 -----+----- 4 -----+----- 5
.....
. mean _b_x _b_cons
Mean estimation                Number of obs    =      10
-----+-----
              |          Mean   Std. Err.   [95% Conf. Interval]
-----+-----
      _b_x |    2.954651   .0384074    2.867767    3.041535
      _b_cons |    .9386876   .0224924    .8878062    .9895689
-----+-----

```


Maximum Likelihood Estimation Method

The CAPM model can be stated as:

$$y_t = \alpha + \beta x_t + u_t \quad (1)$$

where: y_t = Excess return of stock at time t .

x_t = Market excess return at time t .

u_t = Disturbance term at time t .

1. Cauchy Distribution

If the disturbances term u_t are assumed to be Identical Independent Distribution (IID) with Cauchy distribution with density:

$$f(u_t) = (\pi(1+u_t^2))^{-1}$$

To estimate the model using maximum likelihood, we first create program file:

```

program define ml_cauchy
  args lnf Xb
  tempvar res
  quietly gen double `res' = $ML_y1 - `Xb'
  replace `lnf' = ln(1/(_pi*(1+(`res')^2)))
end

. ml model lf ml_cauchy (y=x)

. ml maximize

(240 real changes made)
initial:      log likelihood = -847.98921
(240 real changes made)
(240 real changes made)
alternative:  log likelihood = -834.0182
(240 real changes made)
(240 real changes made)
(240 real changes made)
(240 real changes made)
rescale:     log likelihood = -821.66632
(240 real changes made)
(240 real changes made)
(240 real changes made)
(240 real changes made)
Iteration 0:  log likelihood = -821.66632 (not concave)
(240 real changes made)
(240 real changes made)
(240 real changes made)
(240 real changes made)
(240 real changes made)
(240 real changes made)
(240 real changes made)
(240 real changes made)
(240 real changes made)
(240 real changes made)
(240 real changes made)
(240 real changes made)
(240 real changes made)
(240 real changes made)
(240 real changes made)
(240 real changes made)
Iteration 1:  log likelihood = -625.87317
(240 real changes made)
(240 real changes made)
(240 real changes made)

```

```

(240 real changes made)
(240 real changes made)
(240 real changes made)
Iteration 2: log likelihood = -617.63957
(240 real changes made)
(240 real changes made)
(240 real changes made)
(240 real changes made)
(240 real changes made)
(240 real changes made)
(240 real changes made)
Iteration 3: log likelihood = -617.44101
(240 real changes made)
(240 real changes made)
(240 real changes made)
(240 real changes made)
(240 real changes made)
Iteration 4: log likelihood = -617.44095

```

```

Log likelihood = -617.44095
Number of obs   =      240
wald chi2(1)   =    1455.04
Prob > chi2    =      0.0000

```

	y	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]
	x	.9145708	.0239762	38.14	0.000	.8675783 .9615632
	_cons	.1363996	.1215726	1.12	0.262	-.1018784 .3746776

The default algorithm in estimating ML is Newton-Raphson (nr), if we would like to use BHHH algorithm, we can use the following command.

```

. ml model lf ml_cauchy (y=x), tech(bhhh)
. ml maximize
(240 real changes made)
initial: log likelihood = -847.98921
...
Iteration 5: log likelihood = -617.44095

```

```

Log likelihood = -617.44095
Number of obs   =      240
wald chi2(1)   =    1969.66
Prob > chi2    =      0.0000

```

	y	Coef.	OPG Std. Err.	z	P> z	[95% Conf. Interval]
	x	.914568	.0206073	44.38	0.000	.8741785 .9549575
	_cons	.1364103	.0914454	1.49	0.136	-.0428195 .31564

2. Normal Distribution

If the disturbances term u_t are assumed to be normal distribution. To estimate the model using maximum likelihood, we use the following command:

```
program ml_norm
  version 10
  args todo b lnf
  tempvar thetal sigma
  mlevel `thetal' = `b', eq(1)
  mlevel `sigma' = `b', eq(2)
  tempvar res
  quietly gen double `res' = $ML_y1-`thetal'
  mlsum `lnf' = -0.5*ln(2*_pi)-ln(`sigma')-0.5*((`res'/ `sigma')^ 2)
end
```

```
. ml model d0 ml_norm (y=x) ()
```

```
. ml maximize
```

```
initial:      log likelihood =      -<inf> (could not be evaluated)
feasible:     log likelihood = -13484.369
rescale:      log likelihood = -831.27985
rescale eq:   log likelihood = -761.59207
Iteration 0:  log likelihood = -761.59207 (not concave)
Iteration 1:  log likelihood = -606.96261
Iteration 2:  log likelihood = -593.61441
Iteration 3:  log likelihood = -593.47896
Iteration 4:  log likelihood = -593.47858
Iteration 5:  log likelihood = -593.47858
```

```
Log likelihood = -593.47858      Number of obs      =      240
                                Wald chi2(1)              =      569.99
                                Prob > chi2                =      0.0000
```

	y	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]
eq1	x	.9315372	.039018	23.87	0.000	.8550633 1.008011
	_cons	.1978772	.1878496	1.05	0.292	-.1703013 .5660558
eq2	_cons	2.868786	.1309416	21.91	0.000	2.612145 3.125427

```
. test x=0
```

```
( 1) [eq1]x = 0
```

```
      chi2( 1) = 569.99
      Prob > chi2 = 0.0000
```

```
. est store unres
```

In case of assuming normal distribution, the estimated results are the same as estimating the model using OLS, except the estimated result of standard error of regression (standard deviation of the error term (σ)). In OLS, Root MSE = 2.8808 while in MLE, sigma `_cons` = 2.868786

```
. regress y x
```

Source	SS	df	MS	Number of obs
Model	4691.00669	1	4691.00669	= 240
				F(1, 238) = 565.24
				Prob > F = 0.0000

Residual		1975.18377	238	8.29909148	R-squared	=	0.7037
Total		6666.19046	239	27.8920103	Adj R-squared	=	0.7025
					Root MSE	=	2.8808

	y	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]	
	x	.9315372	.0391816	23.77	0.000	.8543501	1.008724
	_cons	.1978772	.1886373	1.05	0.295	-.1737347	.5694892

```
. ml model d0 ml_norm (y=) ()
```

```
. ml maximize
```

```
initial:      log likelihood =      -<inf> (could not be evaluated)
feasible:    log likelihood = -13484.369
rescale:     log likelihood = -831.27985
rescale eq:  log likelihood = -761.59207
Iteration 0: log likelihood = -761.59207
Iteration 1: log likelihood = -739.46007
Iteration 2: log likelihood = -739.44504
Iteration 3: log likelihood = -739.44504
```

```
Log likelihood = -739.44504
```

	Number of obs	=	240
	wald chi2(0)	=	.
	Prob > chi2	=	.

	y	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]	
eq1	_cons	.9513828	.3401948	2.80	0.005	.2846133	1.618152
eq2	_cons	5.270275	.240554	21.91	0.000	4.798798	5.741752

```
. est store res
```

```
. lrtest unres res
```

```
Likelihood-ratio test
```

(Assumption: res nested in unres)	LR chi2(1)	=	291.93
	Prob > chi2	=	0.0000