Simulation and bootstrapping

CONTENTS:

- **Simulation applications**
- **Bootstrap methods**

Simulation applications

Simulation

A **simulation** is the imitation of the operation of a real-world process or system over time (generation of an artificial history and observation of that observation history). A model constructs a conceptual framework that describes a system. The behavior of the system that evolves over time is studied by developing a simulation model.

Simulation is a powerful pedagogic tool for exposition and illustration of statistical concepts. At the simplest level, we can use (pseudo-)random samples to illustrate distributional features of artificial data.[1]

EXAMPLE 1 (Compare samples generated by two different data generating process (DGP))

The first data generating process is $y_i = 10 + 5x_i + u_i$ where u_i is distributed as a normal with mean 0 and standard deviation 25. The second data generating process is $y_i = 10 + 5x_i + u_i$ where u_i is distributed as a normal with mean 0 and standard deviation 144.

```
. clear

. set seed 12345

. set obs 50
obs was 0, now 50

. gen x=100*uniform()

. gen u=rnormal(0,25)

. gen y=10+5*x+u

. plot y x

 492.041 +
         |                                                    *
         |                                                        * *
         |                                                     ** **
         |                                                  *      *
         |                                               **  ***
         |                                              ***
         |                                          ** *
         |                                     *
         |                                  *  **  *
    y    |
         |
         |                          * * *
         |                      *      *
         |                      *
         |                *    *    *
         |
         |        * *
         | *      * **
         |    ** **
 12.6063 + *   *
         +------------------------------------------------------------------+
         2.16141                         x                          93.343
```

[1] Random sampling refers to the fact that all the individuals in a population have the same probability of being chosen and pseudo random sampling refers to a process that appears to be random but it is not since it is determined by a deterministic causal process.

```
. clear

. set seed 12345

. set obs 50
obs was 0, now 50

. gen x=100*uniform()

. gen u=rnormal(0,144)

. gen y=10+5*x+u

. plot y x

  821.72 +
         |                                                         *
         |
         |
         |
         |                                                              *
         |                                                     *
         |                                       *                     *
         |                                  *  *              *           *
         |                                                  **   *    *
    y    |                     *    *           *           * *   *   *  *
         |               *            *            *    **   *   *        *
         | *                            *                          *
         |                                                 *
         |        * *          *
         |    *   *           *       *
         | *         *
         |     *     *             *
         |       *   *
         |        *
-178.601 +     *
          +----------------------------------------------------------------+
          2.16141                        x                          93.343
```

Since the error term has more variance in the second DGP, we can observe that the data points are more spread.

Monte Carlo

We often want to evaluate the properties of estimators, or compare a proposed estimator to another, in a context where analytical derivation of those properties is not feasible. In that case, econometricians resort to **Monte Carlo studies**: is a simulation exercise design to approximate the sampling distribution of estimators and hence, their properties.

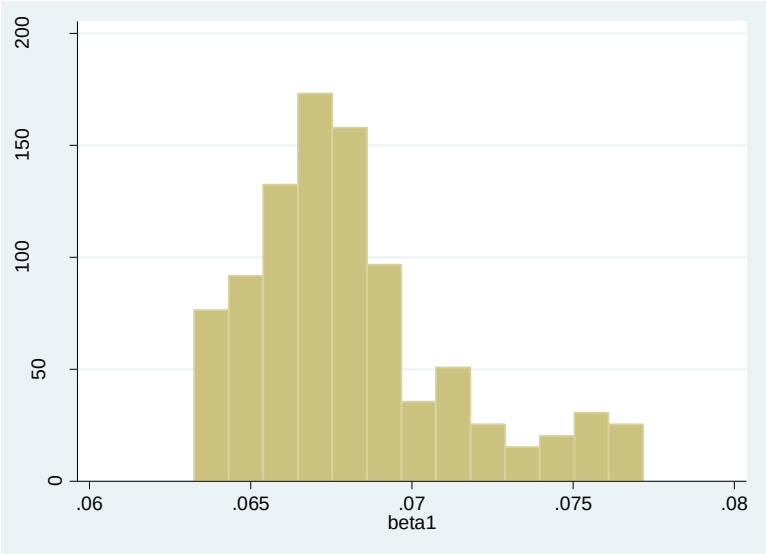
EXAMPLE 2 (Changes in the estimation of a parameter by changing the sample)

This example uses data to estimate the salary of the individuals in logs (*ling*) using as controls the years of schooling (*school*), age (*edad*), and size of the worker's company (dummy variables *tam2* to *tam5*). By replacing a younger individual with an older individual in the sample each time, we obtain different estimates of the impact (coefficient) of schooling on wages. We restrict the analysis to only men (*sexo*=1). The graph represents the distribution of coefficients obtained.

```
. use EJEMPLO4.DTA

. gen ling= log(ingreso)
(2561 missing values generated)

. tab tamanho, g(tam)
```

Tamano de la empresa	Freq.	Percent	Cum.
1	995	32.79	32.79
2	903	29.76	62.56
3	414	13.65	76.20
4	99	3.26	79.47
5	623	20.53	100.00
Total	3,034	100.00	

```
. quietly reg ling school edad tam2 tam3 tam4 tam5 if sexo==1

. gen edad2=edad if e(sample)
(4247 missing values generated)

. sort edad2

. gen orden=_n if e(sample)
(4247 missing values generated)

. mat beta=J(183,1,1)

. quietly for num 1/183: reg ling school edad tam2 tam3 tam4 tam5 if sexo==1 & orden>=X & orden<=X+999 \ mat betaX=e(b) \
> matrix beta[X,1]=betaX[1,1]

. quietly mat list beta

. svmat beta, names (beta)

. histogram beta1
(bin=13, start=.06324627, width=.0010723)
```



Bootstrap methods

A **bootstrap** provides a way to perform statistical inference by resampling from the sample. The statistics being studied are usually standard errors, confidence intervals, or test statistics.

Consider, for example, calculating the standard error of estimator $\hat{\theta}$, which is difficult to perform using conventional methods. Suppose 400 random samples from the population are available. Then, we could obtain 400 different estimates of $\hat{\theta}$ and let the standard error of $\hat{\theta}$ be the standard deviation of these 400 estimates.

In practice, however, only one sample from the population is available. The bootstrap generates multiple samples by resampling from the current sample. Essentially, the observed sample is viewed as the population, and the bootstrap is a method to obtain multiple samples from this population. Given 400 bootstrap resamples, we obtain 400 estimates of $\hat{\theta}$, and then derive the standard error of $\hat{\theta}$.

Let $\hat{\theta}_1^*, \dots, \hat{\theta}_B^*$ denote the estimates, where B=400 in this case. Then, the bootstrap estimate of the variance of $\hat{\theta}$ is

$$\widehat{Var}_{Boot}(\hat{\theta}) = \frac{1}{B-1}\sum_{b=1}^{B}(\hat{\theta}_b^* - \overline{\hat{\theta}^*})^2 \quad (1)$$

where $\overline{\hat{\theta}^*} = \frac{1}{B}\sum_{b=1}^{B}\hat{\theta}_b^*$ is the average of the B bootstrap estimates. The square root of $\widehat{Var}_{Boot}(\hat{\theta})$, denoted by $se_{Boot}(\hat{\theta})$, is called the bootstrap estimate of the standard error of $\hat{\theta}$.

A general **bootstrap algorithm** is defined follows:

1. Given the data $\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_N$, draw a bootstrap sample of size N (using any bootstrap sampling method) and denote this new sample $\mathbf{w}_1^*, \mathbf{w}_2^*, \dots, \mathbf{w}_N^*$.

2. Calculate the corresponding statistic(s) using the bootstrap sample. Examples include:
   - The estimate $\hat{\theta}^*$ of $\hat{\theta}$
   - The standard error $s_{\hat{\theta}^*}$ of the estimate $\hat{\theta}^*$
   - The t-statistic $t^* = (\hat{\theta}^* - \hat{\theta})/s_{\hat{\theta}^*}$ centered at the original estimate $\hat{\theta}$.

   Hence, $\hat{\theta}^*$ and $s_{\hat{\theta}^*}$ are calculated in the usual way but using the new bootstrap sample rather than the original sample.

3. Repeat steps 1 and 2 B independent times, where B is a large number (~300), obtaining B bootstrap replications of the statistics of interest, such as $\hat{\theta}_1^*, \dots, \hat{\theta}_B^*$ or $t_1^*, \dots, t_B^*$.

4. Use these B bootstrap replications to obtain a bootstrapped version of the statistic.

We can use different **bootstrap sampling methods**:

Nonparametric bootstrap or **paired bootstrap:** Obtain $\mathbf{w}_1^*, \mathbf{w}_2^*, \dots, \mathbf{w}_N^*$ by sampling with replacement from $\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_N$. It is called paired bootstrap since in single-equation regression models $\mathbf{w}_i = (y_i, \mathbf{x}_i)$, so here both y_i and $\mathbf{x}_i$ are resampled.

Parametric bootstrap: Suppose the conditional distribution of the data is specified, say $y|\mathbf{x}\sim F(\mathbf{x}, \boldsymbol{\theta}_0)$ and an estimate $\widehat{\boldsymbol{\theta}} \xrightarrow{p} \boldsymbol{\theta}_0$ is available. We can obtain a bootstrap sample by using the original $\mathbf{x}_i$ while generating y_i by random draws from F ($\mathbf{x}_i, \widehat{\boldsymbol{\theta}}$).

Residual bootstrap: For regression model with additive iid errors, say $y_i = g(\mathbf{x}_i, \beta) + u_i$, we can form fitted residuals $\hat{u}_1, \ldots, \hat{u}_N$, where $\hat{u}_i = y_i - g(\mathbf{x}_i, \hat{\beta})$. Bootstrap from these residuals to get a new draw of residuals, say ($\hat{u}_1^*, \ldots, \hat{u}_N^*$) leading to a bootstrap sample ($y_1^*, \mathbf{x}_1$),..., ($y_N^*, \mathbf{x}_N$) where $y_1^* = g(\mathbf{x}_i, \hat{\beta}) + u_i^*$.

EXAMPLE 3 (Bootstrapped standard errors)

Stata has two options for bootstrapping. Most model estimation commands have a *vce(bootstrap)* option for estimating coefficient standard errors and a *bootstrap* command useful to bootstrap more complex expressions. When used for the same purpose the results are the same. Note that the option *vce(bootstrap)* uses paired bootstrap as a sampling method.

Using the data in housing prices (HPRICE1.RAW) from the chapter in Linear Models, we can compare the bootstrapping standard errors with the standard errors without correction. Remember that after performing the Breusch-Pagan /Cook-Weisberg test we concluded that heteroskedasticity was present in this data set.

As can be seen, without correcting for heteroskedasticity we reject the null hypothesis that *lotsize* is equal to zero. Once we use bootstrapped standard errors we can see that the point estimated does not change but standard errors do. The coefficient of *lotsize* becomes statistically insignificant (same when we use the option 'r' in regress).

```
. use http://fmwww.bc.edu/ec-p/data/wooldridge/hprice1

. regress price lotsize sqrft bdrms
```

Source	SS	df	MS		
Model	617130.701	3	205710.234	Number of obs	= 88
Residual	300723.805	84	3580.0453	F(3, 84)	= 57.46
				Prob > F	= 0.0000
Total	917854.506	87	10550.0518	R-squared	= 0.6724
				Adj R-squared	= 0.6607
				Root MSE	= 59.833

price	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
lotsize	.0020677	.0006421	3.22	0.002	.0007908	.0033446
sqrft	.1227782	.0132374	9.28	0.000	.0964541	.1491022
bdrms	13.85252	9.010145	1.54	0.128	-4.065141	31.77018
_cons	-21.77031	29.47504	-0.74	0.462	-80.38466	36.84405

Using bootstrap standard errors:

```
. regress price lotsize sqrft bdrms,vce(bootstrap, reps (500))
(running regress on estimation sample)

Bootstrap replications (500)
----+--- 1 ---+--- 2 ---+--- 3 ---+--- 4 ---+--- 5
..................................................    50
..................................................   100
..................................................   150
..................................................   200
..................................................   250
..................................................   300
..................................................   350
..................................................   400
..................................................   450
..................................................   500

Linear regression                               Number of obs     =         88
                                                Replications      =        500
                                                Wald chi2(3)      =      78.12
                                                Prob > chi2       =     0.0000
                                                R-squared         =     0.6724
                                                Adj R-squared     =     0.6607
                                                Root MSE          =    59.8335

------------------------------------------------------------------------------
             |   Observed   Bootstrap                         Normal-based
       price |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
     lotsize |   .0020677   .0037322     0.55   0.580    -.0052473    .0093827
       sqrft |   .1227782   .0254457     4.83   0.000     .0729054    .1726509
       bdrms |   13.85252   9.553663     1.45   0.147    -4.872314    32.57736
       _cons |  -21.77031   35.44406    -0.61   0.539    -91.23938    47.69876
------------------------------------------------------------------------------
```

EXERCISE 1 (Central Limit Theorem)

a) State in your own words the Central Limit Theorem (CLT)

Now, we will use Stata to generate 10,000 random samples according to a DGP that we specify. In each random sample, we will calculate the sample mean $\bar{x}$ and sample standard deviation $\hat{\sigma}$. After we have collected the results, we will have a new Stata dataset, consisting of 10,000 observations, where each observation has a $\bar{x}$ and a $\hat{\sigma}$. We can then look at the distribution of $\bar{x}$.

b) Write the following commands in a do file and run it three times setting the number of repetitions to 100, 1,000 and 10,000. Save the resulting graph each time and compare. Comment on the results following the logic behind the CLT.

```
capture program drop mysim
program define mysim , rclass
drop _all
set obs $obs
gen x = rnormal(1,2)
sum x
return scalar m = r(mean)
return scalar sd = r(sd)
end

set more off
global obs 50
simulate mean=r(m) stdev=r(sd), reps(#) : mysim
hist mean , normal

bro mean stdev
```

c) Why do we use the *simulate* command? Can you think any other way to do the same exercise without using *simulate*? (Find information about simulate typing *help simulate*)