Exercise – Understanding the sampling distribution of the OLS estimator using simulation

1. Aim of the exercise

Suppose you estimate a simple linear regression model in standard econometric software like Stata. In the regression output, among other statistics, you find an OLS coefficient estimate and its standard error. The coefficient estimate presented is just one number. So why is there a standard error associated with this single number? The reason is that the OLS estimator is a random variable and therefore has a distribution, even though you don't observe this distribution directly. However, you can simulate it through a conceptual experiment. In this exercise, we will simulate this distribution. Understanding why an OLS coefficient estimate has a standard error is important also to understand various key concepts in econometrics. For instance, studying the statistical properties of an estimator hinges on the fact that it has a distribution.

2. Create the sampling distribution of the OLS estimator

2.1. Clear the memory

Clear the memory from possible calculations from an earlier session.

```
8 % 2.1. Clear the memory
9 clear;
```

2.2. Set the sample size

Assume that the model of interest is a simple linear regression model that includes a constant term and an independent variable. Assume that there are `N_obs` observations available for this independent variable and for the dependent variable of the regression.

```
11 % 2.2. Set the sample size
12 N_obs = 9000;
```

2.3. Generate data for the independent variable

Create the constant term. The code presented in this section draws `N_obs` theoretical observations from the uniform distribution to create an artificial dataset for the independent variable X. It is not important which distribution we use. In our simulation exercise below, we will use these observations to generate data for y repeatedly. In the exercise we will keep the observations of X fixed. That is, as we will mimic taking repeated samples from the population, we will be doing this only for y and not for X. This means that we will keep the random data of X fixed in repeated sampling. Keeping X fixed in repeated sampling is indeed an assumption we make. Note, however, that this is the classical assumption we make while we derive the basic econometric theory. That is, we condition on the values of a regressor while we make econometric derivations. We do this because it simplifies the derivations, and the basics of econometric theory does not change. In line 17 we create the systematic component of the regression equation.

```
14 % 2.3. Generate data for the independent variables
```

```
x_0 = ones(N_obs,1);
x_1 = random('Uniform',-1,1,[N_obs 1]);
X = [x_0 x_1];
```

2.4. Set (hypothetical) values for the population coefficients of the regression model

Assume hypothetical values for the population coefficients of the independent variables. We need this to generate values for y later in the code. In principle we do not observe the population (also called true) coefficients.

```
% 2.4. Set (hypothesized) values for the population coefficients
B_true_0 = 0.2;
B_true_1 = 0.5;
B_true = [B_true_0; B_true_1];
```

2.5. Set the number of simulations

In this exercise a simulation refers to taking a random sample from the population. Since we want to take samples from the population repeatedly, we will be repeating the simulation multiple times. Here we define the number of simulations or samples.

```
% 2.5. Set the number of simulations
N_sim = 1000;
```

2.6. Preallocate vectors for storing OLS estimates from all samples

Create empty vectors that will store simulated coefficient estimates. The dimension of a vector is `1` $\times$ `N_sim` because for a given coefficient we will have `N_sim` coefficient estimates from repeated samples from the population.

```
% 2.6. Preallocate a matrix for storing OLS estimates from all samples
B_hat_0_sim = NaN(1,N_sim);
B_hat_1_sim = NaN(1,N_sim);
```

2.7. Preallocate vectors for storing the standard error estimates

Create empty vectors that will store `N_sim` standard error estimates for the two OLS coefficient estimates from repeated samples. Remember that there is a standard error estimate for each coefficient estimate.

```
% 2.7. Preallocate a matrix for storing the standard error estimates
% (SSEs) from repeated samples
B_hat_0_SEE_sim = NaN(1,N_sim);
B_hat_1_SEE_sim = NaN(1,N_sim);
```

2.8. Create the sampling distribution of the OLS estimator

The aim of this exercise is to make an educated guess of the distribution of the OLS estimate of the coefficient of `x_1`. In the for loop considered in this section, we pretend that we are drawing `N_sim` random samples from the population. Each sample leads to an estimate of the coefficient of `x_1`. This leads to a distribution for this OLS estimate.

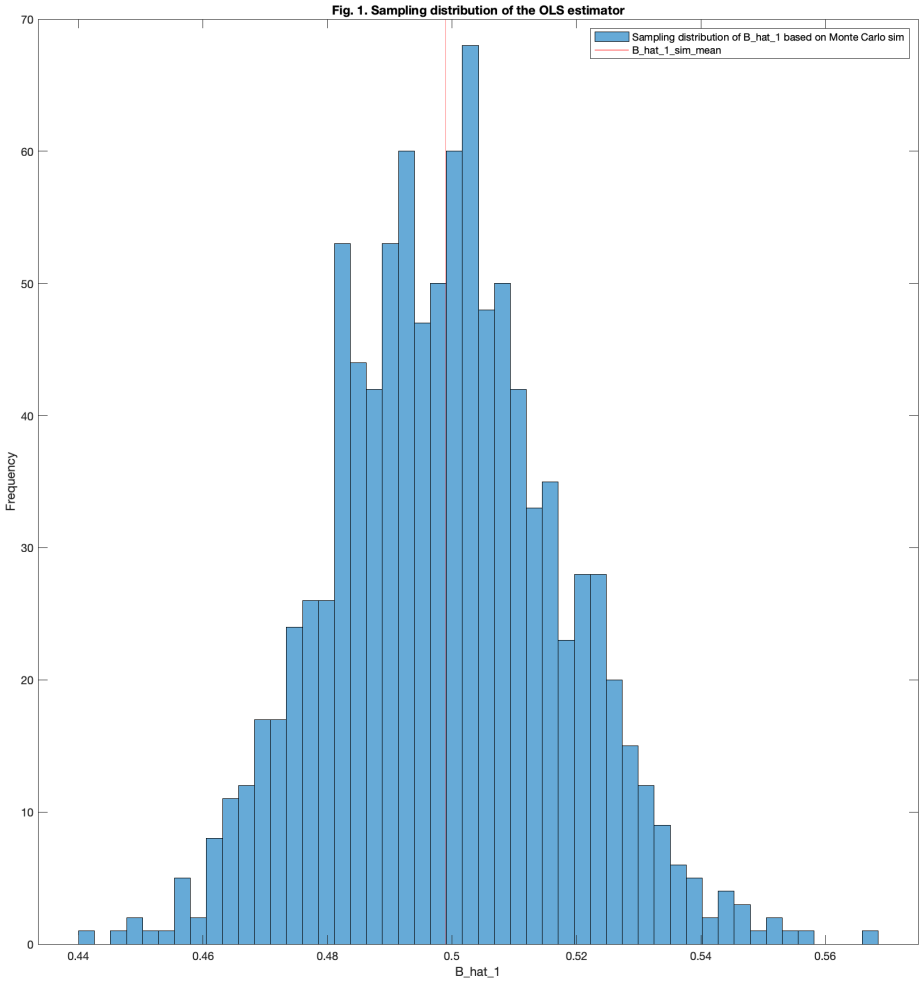
The for loop carries out the simulation. Line 37 is the index of the for loop that instructs the for loop to execute a program, still to be specified, `N_sim` times. Line 39 draws random values from the standard normal distribution for the error term of the regression that is of the same dimension of the dependent variable which is `N_obs` $\times$ `1`. In line 41, using the generated data for the independent variables, the true values for the population coefficients, and the generated values for the error term, we generate new data for the dependent variable at each iteration of the for loop. This gives the true data generating process (DGP). Using the DGP, we obtain samples from "repeated sampling", or "sampling in the long run". In line 43, we estimate the regression equation using an external function that accepts the generated data for `y` and `X` and returns standard OLS statistics as output. Lines 45 and 46 store the new coefficient estimates in `B_hat_sim(:,i)` at iteration `i` of the for loop. Lines 48 and 49 do this for the standard error estimators. In the last line, `end` marks the end of the for loop. What we have just carried out is a Monte Carlo simulation.

```
36 % 2.8. Create the sampling distribution of the OLS estimator
37 for i = 1:N_sim
38     % Draw new error for each sample (in each iteration of the loop)
39     u = random('Normal',0,1,[N_obs 1]);
40     % Generate values for the dependent variable
41     y = X*B_true+u; % The data generating process (DGP)
42     % Obtain OLS statistics using the external function
43     LSS = exercisefunctionlss(y,X);
44     % Store the OLS estimates
45     B_hat_0_sim(1,i) = LSS.B_hat(1,1);
46     B_hat_1_sim(1,i) = LSS.B_hat(2,1);
47     % Store the SE estimates of OLS estimates
48     B_hat_0_SEE_sim(1,i) = LSS.B_hat_SEE(1,1);
49     B_hat_1_SEE_sim(1,i) = LSS.B_hat_SEE(2,1);
50 end
```

3. Plot the sampling distribution of the OLS estimator

Here we plot the simulated estimates of the coefficient of the independent variable. This is the sampling distribution of this OLS estimate.

```
52 %% 3. Plot the sampling distribution of the OLS estimator
53 figure
54 hold on
55 histogram(B_hat_1_sim(1,:),50)
56 line([mean(B_hat_1_sim(1,:)) mean(B_hat_1_sim(1,:))],ylim,'Color','red')
57 title('Fig. 1. Sampling distribution of the OLS estimator')
58 legend('Sampling distribution of B\_hat\_1 based on Monte Carlo sim',...
59     'B\_hat\_1\_sim\_mean')
60 ylabel('Frequency')
61 xlabel('B\_hat\_1')
62 hold off
```

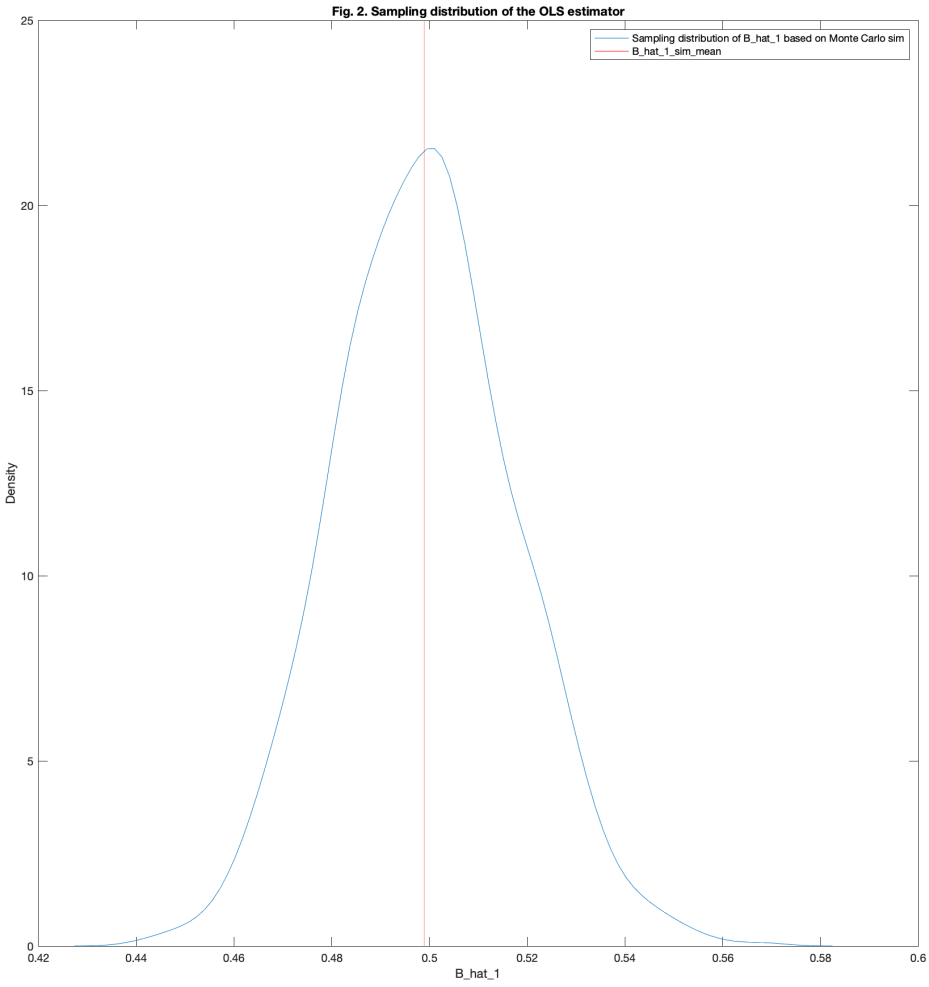


4. Plot the sampling distribution of the OLS estimator as a density

Here we use the built-in MATLAB function `ksdensity` to plot a smoothed version of the frequency distribution produced in the preceding section. This is to better visualize the sampling distribution.

```
%% 4. Plot the sampling distribution of the OLS estimator as a density
figure
hold on
ksdensity(B_hat_1_sim(1,:))
line([mean(B_hat_1_sim(1,:)) mean(B_hat_1_sim(1,:))],ylim,'Color','red')
title('Fig. 2. Sampling distribution of the OLS estimator')
legend('Sampling distribution of B\_hat\_1 based on Monte Carlo sim',...
    'B\_hat\_1\_sim\_mean')
ylabel('Density')
```

```
xlabel('B\_hat\_1')
hold off
```



5. Mean of the sampling distribution of the OLS estimator

Calculate the mean of the sampling distribution of the OLS estimator of `mean(x_1)` using the command `mean(B_hat_1_sim)`. Observe that the mean is close to the true value that we assumed for `B_true_1` which is 0.5. This should not surprise us. The OLS estimator of a coefficient is an unbiased estimator of the true value of that coefficient.

```
%% 5. The mean of the sampling distribution of the OLS estimator
mean(B_hat_1_sim)
```

6. Standard error of a statistic is the standard deviation of its sampling distribution

Why does a given coefficient estimate have a variance? It is just one number after all. The

answer is that we are thinking of a conceptual experiment. The conceptual experiment is that we take random samples from the population, in each sample we calculate a coefficient estimate, and create a distribution for this estimate. This is the sampling distribution of the OLS estimator. Hence, the OLS estimator has a mean and a variance. A given coefficient estimate is a random variable, and hence has a distribution. It is just that we do not observe this distribution because we are not able to repeatedly take samples from the population, to calculate a coefficient estimate in each sample, and plot its sampling distribution.

We do not know observe the distribution of a coefficient estimate but we still want to know about it. In particular, we want to know about the variance of the coefficient estimate because we want a measure of how close this estimate is to the true, population coefficient. Therefore, we use an estimator, an analytical formula, for this variance. This estimator is given by `LSS.B_hat_VCE` in the external function file `exercisefunction.m`. The suffix `VCE` stands for the 'variance-covariance estimator'. The square root of it is the standard error estimator of the OLS estimator.

We just argued that we do not observe the distribution of the OLS estimator. However, in Section 4, using simulation, we created an approximate distribution for the OLS estimator based on some hypothetical data and regression model. We simulated the sampling distribution of the OLS estimator. This distribution gives us an intuitive understanding of what a standard error represents. The standard error of a statistic is the standard deviation of its sampling distribution: https://en.wikipedia.org/wiki/Standard_error. The OLS estimator is a statistic. Hence, the estimator of the standard error of the OLS estimator should be equal to the standard deviation of the sampling distribution of the OLS estimator created in Section 4. The code presented at the end of this section confirms this. In particular, first note that `B_hat_1_sim(1,:)` gives the sampling distribution of the OLS estimator of the population coefficient of `x_1`. `std(B_hat_1_sim(1,:))` gives the standard deviation of this sampling distribution. Second note that the estimated standard error of the OLS estimator of the population coefficient of `x_1` is given by `LSS.B_hat_SEE(2,1)`. `LSS.B_hat_SEE(2,1)` should be very close to `std(B_hat_1_sim(1,:))`. In fact they are. The small difference is due to simulation noise.

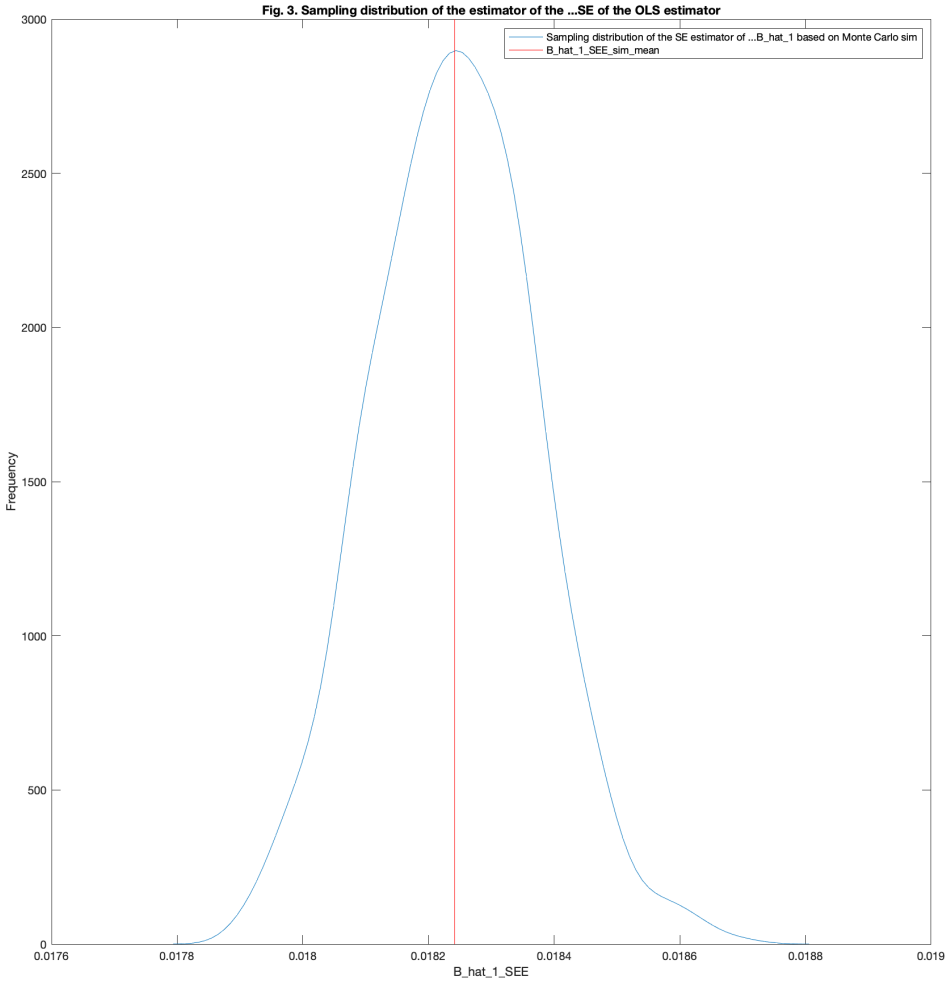
```
79 %% 6. SE of a statistic is the SD of its sampling distribution
80 std(B_hat_1_sim(1,:))
81 LSS.B_hat_SEE(2,1)
```

7. Plot the sampling distribution of the standard error estimator

The vector `B_hat_1_SEE_sim(1,:)` contains `N_sim` simulated standard error estimates for the `N_sim` simulated estimates of the coefficient of `x_1`. Plot the sampling distribution of the standard error estimator. Notice that we have a sampling distribution because the standard error estimator is a random variable just like that the OLS estimator is a random variable. From one sample to another, the random variable takes different realizations.

```
83 %% 7. Plot the sampling distribution of the SE estimator
84 figure
85 hold on
86 ksdensity(B_hat_1_SEE_sim(1,:))
87 line([mean(B_hat_1_SEE_sim(1,:)) mean(B_hat_1_SEE_sim(1,:))],ylim,...
88     'Color','red')
89 title(['Fig. 3. Sampling distribution of the estimator of the ...' ...
90     'SE of the OLS estimator'])
```

```
legend(['Sampling distribution of the SE estimator of ...' ...
    'B\_hat\_1 based on Monte Carlo sim'],'B\_hat\_1\_SEE\_sim\_mean')
ylabel('Frequency')
xlabel('B\_hat\_1\_SEE')
hold off
```



8. Final notes

This file is prepared and copyrighted by Tunga Kantarcı. This file and the accompanying MATLAB files are available on GitHub. They can be accessed via this link.