Empirical exercise – Sampling distribution of the OLS estimator

1. Aim of the exercise

Suppose that you estimate a simple linear regression model in standard econometric software such as Stata. In the regression output, typically presented, among other statistics, is an OLS coefficient estimate, and the standard error of this estimate. The presented coefficient estimate is just one number. How come that there is a standard error for this one number? Is not the standard error a matter of a random variable? The answer is that the OLS estimator is a random variable. It has a distribution. It is just that you do not observe it. But you can simulate it in a conceptual experiment. In this exercise we will simulate this distribution. We do not do this only to understand why an OLS coefficient estimate has a standard error. Learning about this distribution is key to understanding many other key concepts in econometrics. For example, it is because the OLS estimator, or any other estimator, has a distribution that you study the statistical properties of that estimator.

2. Set a seed for reproducible results

In this exercise we will consider a simulation exercise where we will draw random numbers using a random number generator. As this will get clear once we finish the exercise, we will happen to want to hold the random numbers generated fixed every time we re-initiate the random number generator. This will allow us to inspect the changes in the results when we change a certain parameter so that we know that the change in the results is not due to the random numbers re-generated but due to changing the value of the parameter of interest. `rng` is a built-in MATLAB function that sets a seed for the random numbers generated for reproducible results.

```
clear;
rng(1)
```

3. Set the number of simulations

Define the number of simulations to be carried out. In the current exercise, the number of simulations is referring to the number of random samples we will draw from a given distribution. This is a simulation because we imitate randomly drawing samples from the population.

```
N_sim = 1000;
```

4. 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 also that we have `N_obs` observations for each variable of the model.

```
N_obs = 9000;
```

5. Set true values for the coefficients of the intercept and the independent variable

Assume that we know the true values of the coefficients of the variables of the simple linear regression model. For the simulation exercise we will carry out, it is not important whether or not we know the true values. We could as well have some sample data, estimate the coefficients based on this data, and use them, and data based on repeated samples from the sample data, in the simulation.

```
B_true = [0.2; 0.5];
N_par = 2;
```

6. Generate data for the independent variable

Create the constant term. Draw a set of random numbers from the uniform distribution, and require these numbers to be in the range $[-1, 1]$. Consider this vector as the (only) independent variable of the regression model. It is not important if we simulate the data for the independent variable. We could as well have some sample data for the independent variable, and use it in the simulation. It is also not important from which distribution we draw the random numbers for the independent variable. Create the systematic component of the regression equation, and name it as `X`.

```
x_0 = ones(N_obs,1);
x_1 = unifrnd(-1,1,N_obs,1);
X = [x_0 x_1];
```

7. Create an empty matrix for storing the simulated coefficient estimates

Create an empty matrix that will store simulated coefficient estimates. The dimension of the matrix is `N_par` $\times$ `N_sim` because we have `N_par` coefficients to estimate, and `N_sim` coefficient estimates to simulate. Also, create an empty matrix that will store simulated standard error estimates of the OLS coefficient estimates.

```
B_hat_sim = NaN(N_par,N_sim);
B_hat_SEE_sim = NaN(N_par,N_sim);
```

8. Create a sampling distribution for 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 at the end of the 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.

Presented at the end of the section is a for loop that carries out the simulation. The first line of the for loop is the index of the for loop that instructs the for loop to execute a program, still to be specified, `N_sim` times.

In the second line of the for loop, we draw 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`.

Using 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 true DGP, we obtain data in 'repeated sampling', or 'sampling in the long run'.

In the fourth line of the for loop, we estimate the regression equation using the data generated for `y` and `X`. In lines five and six, we then instruct MATLAB to store the new coefficient estimates in `B_hat_sim(:,i)` at iteration `i` of the for loop. `end` marks the end of the for loop.

What we have just carried out is a Monte Carlo simulation. A similar type of simulation could be used to obtain an approximation of the covariance matrix of the coefficient estimates.

```
for i = 1:N_sim;
    u = normrnd(0,1,N_obs,1);
    y = X*B_true+u;
    LSS = exercisefunction(y,X);
    B_hat_sim(1,i) = LSS.B_hat(1,1);
    B_hat_sim(2,i) = LSS.B_hat(2,1);
    B_hat_SEE_sim(1,i) = LSS.B_hat_SEE(1,1);
    B_hat_SEE_sim(2,i) = LSS.B_hat_SEE(2,1);
end
```

9. Plot the sampling distribution of the OLS estimator of the coefficient of the independent variable

Plot the simulated estimates of the coefficient of the independent variable using the command at the end of the section. This is the sampling distribution of the OLS estimator of the coefficient of the independent variable.

```
histogram(B_hat_sim(2,:),50)
hold on
line([mean(B_hat_sim(2,:))  mean(B_hat_sim(2,:))],ylim,'Color','red')
hold off
title('Figure 1:  Sampling distribution of the OLS estimator')
legend('Sampling distribution of B_hat_1 based on Monte Carlo sim','B_hat_1_sim_mean')
ylabel('Frequency')
xlabel('B_hat_x_1')
```

10. Plot the sampling distribution of the OLS estimator of the coefficient of the independent variable as a density

The code presented at the end of the section plots a smoothed version of the frequency distribution produced in Section 9 using the built-in MATLAB function `ksdensity`. This is just to better visualise the sampling distribution constructed above.

```
ksdensity(B_hat_sim(2,:))
hold on
line([mean(B_hat_sim(2,:))  mean(B_hat_sim(2,:))],ylim,'Color','red')
hold off
title('Figure 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')
```

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

Why does a given coefficient estimate has 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 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 take samples from the population, to calculate a coefficient estimate in each sample, and plot a sampling distribution.

We do not know observe the distribution of a coefficient estimate but we still want to know about it, and 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 coefficient. Therefore, we use an estimator for this variance. This estimator is given by `LSS.B_hat_VCE` in the 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 8, 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. 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 8. The code presented at the end of the section confirms this. In particular, first note that `B_hat_sim(2,:)` gives the sampling distribution of the OLS estimator of the population coefficient of `x_1`. `std(B_hat_sim(2,:))` gives the standard deviation of this sampling distribution. 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_sim(2,:))`. In fact they are. The small difference is due to simulation noise.

```
std(B_hat_sim(2,:))
LSS.B_hat_SEE(2,1)
```

12. Plot the sampling distribution of the standard error estimator

The vector `B_hat_SEE_sim(2,:)` 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 realisations.

```
ksdensity(B_hat_SEE_sim(2,:))
hold on
line([mean(B_hat_SEE_sim(2,:))  mean(B_hat_SEE_sim(2,:))],ylim,'Color','red')
hold off
title('Figure 3:  Sampling distribution of the estimator of the standard error of
the OLS estimator')
legend('Sampling distribution of the standard error estimator of B_hat_1 based on
Monte Carlo sim','B_hat_1_SEE_sim_mean')
ylabel('Frequency')
xlabel('B_hat_1_SEE')
```