Empirical exercise – Sampling distribution of the OLS estimator and sample size

1. Set a seed for reproducible results

Set a seed for reproducible results.

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

2. Set the number of simulations

Set the number of simulations to be carried out.

```
N_sim = 1000;
```

3. Set the sample size

Assume that we have a linear regression model that contains a constant term and an independent variable. Assume also that we have `N_obs` observations for the variables of this model.

```
N_obs = [1000 10000 100000];
N_obs_j = size(N_obs,2);
```

4. 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 linear regression model we consider, and that these values are as indicated at the end of the section.

```
B_true = [0.2; 0.5];
N_par = 1;
```

5. Create an empty matrix for storing the simulated OLS coefficient estimates

The code presented at the end of the section creates an empty vector, and an empty matrix. The empty vector is `N_sim` × `N_par` because the vector is to store `N_sim` coefficient estimates of the only independent variable `x_1` from a given simulation using a certain number of observations (sample size). The empty matrix is to store in each of its columns the `N_sim` simulated coefficient estimates from three different scenarios featuring different numbers of observations. Therefore the matrix is `N_sim` × `N_obs_j`, where `N_obs_j` is the number of different scenarios of numbers of observations.

```
B_hat_sim_x_1 = NaN(N_sim,N_par);
```

```
B_hat_sim_x_1_j = NaN(N_sim,N_obs_j);
```

6. Create sampling distributions for the OLS coefficient estimates using different sample sizes

Convince yourself that the for loop presented at the end of the section creates three different sampling distributions for the OLS coefficient estimates using three different sample sizes.

A note for avoiding a computational hurdle is the following. The presented for loop makes use of the user-written function `exercisefunctionlss`. The function calculates a set of OLS statistics. However, the for loop used here only needs the OLS coefficient estimates, and therefore other OLS statistics need not be calculated. To avoid an unnecessary waiting time for the for loop to finish its iterations, go to the function file, and mask the code except the part of it calculating the OLS coefficient estimates. You can mask the code by selecting the code, and then by pressing the ‘Comment’ button located on the toolbar of the Editor of your open script file.

```
for j = 1:N_obs_j
    for i = 1:N_sim
        u = normrnd(0,1,N_obs(1,j),1);
        x_0 = ones(N_obs(1,j),1);
        x_1 = unifrnd(-1,1,N_obs(1,j),1);
        X = [x_0 x_1];
        y = X*B_true+u;
        LSS = exercisefunctionlss(y,X);
        B_hat_sim_x_1(i,1) = LSS.B_hat(2,1);
    end
    B_hat_sim_x_1_j(:,j) = B_hat_sim_x_1(:,1);
end
```

7. Plot the sampling distributions of the OLS coefficient estimates at different sample sizes

The code presented at the end of the section plots three different sampling distributions featuring three different sample sizes. It demonstrates the consistency property of the OLS estimator.

Note that this exercise shows consistency of the estimator without doing any theoretical derivation such as taking the probability limit of the estimator. Indeed, when it is not possible to theoretically prove the consistency, or some other property, of an estimator, one can rely on a simulation study, as done here, to investigate the properties of the estimator of interest.

```
ksdensity(B_hat_sim_x_1_j(:,1))
hold on
ksdensity(B_hat_sim_x_1_j(:,2))
hold on
ksdensity(B_hat_sim_x_1_j(:,3))
title('Sampling distribution of the OLS estimator and sample size')
legend('N_obs = 1,000','N_obs = 10,000','N_obs = 100,000')
ylabel('Frequency')
```

```
xlabel('B_hat_x_1')
```