Empirical exercise – Violation of the zero conditional mean due to omitting a variable

1. Aim of the exercise

Omitting an independent variable from the regression equation can violate the zero conditional mean assumption. In this exercise we study the implications of violating this assumption for the sampling distribution of the OLS estimator using simulation.

2. Set a seed for reproducible results

Set a seed for reproducible results.

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

3. Set the number of simulations

Set the number of simulations to be carried out.

```
N_sim = 1000;
```

4. Set the sample size

Assume a linear regression model with `N_obs` observations available for the variables of this model.

```
N_obs = 1000;
```

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

Assume that the liner regression model contains a constant term and two independent variables. Assume also that this is the true DGP and that we know the true values of the coefficients of the variables of this model.

```
B_true = [0.2 0.5 0.75]';
```

6. Create the constant term

Create the constant term.

```
x_0 = ones(N_obs,1);
```

7. Create a vector of covariances between two independent variables

A problem related to the systematic part of the DGP is omitting a relevant independent variable that is part of the true DGP. While this might stem from different reasons, one potential reason is when a researcher does not know that a particular variable belongs in the specification or cannot collect data on that variable. If the omitted independent variable is uncorrelated with all of the independent variables that are included in the regression model, leaving it out will usually have only a small effect on model efficiency. However, if the omitted independent variable is correlated with one or more of the independent variables that are included in the model, this will bias the coefficient estimates.

In this exercise, we examine the omitted variable problem across a range of correlations between the included and omitted independent variables. Let us consider two independent variables, correlated at 11 different values from 0 to 0.99. The first line of the code at the end of the section creates a vector array containing the different correlation values. We also consider these values as covariances by assuming that each of the two correlated variables has a variance of 1. The second line of the code assigns the size of the column dimension of this vector to a variable named `covariance_par_j`. We will use this variable when iterating over different correlation scenarios in our simulation.

```
covariance_par = [0:0.1:0.9 0.99];
covariance_par_j = size(covariance_par,2);
```

8. Create an empty matrix to store the simulated OLS coefficient estimates

Create an empty matrix that will store the coefficient estimates simulated at different levels of correlation between the included and omitted independent variables. The matrix is `N_sim` $\times$ `covariance_par_j` because we have `N_sim` coefficients to simulate, and `covariance_par_j` different levels of correlation between the included and omitted independent variable.

```
B_hat_x_1_sim = NaN(N_sim,covariance_par_j);
```

9. Define input arguments for the multivariate normal random number generator

In the next section, we will draw random numbers from the multivariate normal distribution to create two independent variables that are correlated with each other. To do this, we will make use of the built-in MATLAB function `mvnrnd`. The function accepts three input arguments. The first input argument is the mean vector of the distribution. The second input argument is the covariance matrix of the distribution. The third input argument specifies the number of observations to be drawn for each random variable of the distribution. Here we define the first and the third input arguments. The second input argument is to be defined within the simulation in the next section because in each iteration of the simulation the covariance matrix will be updated in accordance with the different levels of correlation between two independent variables.

```
MU = [0 0];
cases = N_obs;
```

10. Create sampling distributions for the OLS coefficient estimates under different correlation levels between the included and omitted independent variables

At the end of the section we consider two for loops. The inner for loop simulates `N_sim` coefficient estimates from repeated sampling. The outer for loop repeats this simulation for 11 different correlation levels between the included and the omitted variables.

Consider the inner for loop. In the first line of the for loop we define the index of the for loop. In the second line we define the covariance matrix of the included and omitted variables at a given covariance value taken from the range of values contained in the vector array `covariance_par`. We set the variances of each independent variable to 1. In the third line, we supply the `mvnrnd` function with the defined covariance matrix, and with two other input arguments defined in the previous section. The function generates random values for the included and omitted variables from the multivariate normal distribution so that the variables are correlated. In the fourth and fifth lines of the for loop we define the included (`x_1`) and omitted (`x_2`) variables. In line six we generate random values for the error term. Note that we rule out heteroskedasticity by setting the standard deviation of the error term to 1. In line seven, we construct the systematic component of the regression equation. In line eight, we generate values for the dependent variable. In line nine, we define the systematic component of a new regression equation that omits variable `x_2`. We then use the function `exercisefunction` to estimate the coefficient of `x_1` in this regression. We collect the simulated coefficient estimates at different levels of correlation between the included and the omitted variable in the matrix array `B_hat_x_1_sim(i,j)`.

```
for j = 1:covariance_par_j
    for i = 1:N_sim
        SIGMA = reshape([1 covariance_par(:,j) covariance_par(:,j) 1],2,2);
        x_1_x_2_correlated = mvnrnd(MU,SIGMA,cases);
        x_1 = x_1_x_2_correlated(:,1);
        x_2 = x_1_x_2_correlated(:,2);
        u = normrnd(0,1,N_obs,1);
        X_with_x_2 = [x_0 x_1 x_2];
        y = X_with_x_2*B_true+u;
        X_without_x_2 = [x_0 x_1];
        LSS = exercisefunctionlss(y,X_without_x_2);
        B_hat_x_1_sim(i,j) = LSS.B_hat(2,1);
    end
end
```

11. Plot the sampling distributions of the OLS coefficient estimates of the included independent variable under different correlation levels between the included and omitted independent variables

The plot at the end of the section gives the density estimate of the 1000 coefficient estimates of `x_1`, at a correlation of 0 and 0.99 between `x_1` and `x_2`. The distribution of estimates at correlation 0 is centered right at 0.5, indicating no bias. This demonstrates that omitting a variable that is not correlated with an included variable does not affect parameter estimates in the case of OLS (though this is not true for all estimators). In contrast, the distribution of estimates when the correlation is 0.99 shows a considerable amount of bias. In fact, recall from the code in a previous section that the coefficient on the omitted variable is set to 0.75. The mean of the distribution with correlation 0.99 is 1.242, which is 0.5 (true coefficient of `x_1`) plus 0.742. Hence, at near-perfect correlation with the omitted variable, almost all of the true effect of that omitted variable is incorrectly attributed to `x_1` through the biased estimate of the coefficient of `x_1`.

You can explore how changing aspects of this simulation changes the nature of the results. What if there is a third variable that is included in the model but it is not correlated with the omitted variable? What if you increase the sample size for each draw in the simulation? These kinds of exploration will help you see the real nature of your DGP and how your chosen statistical estimator performs.

```
ksdensity(B_hat_x_1_sim(:,1))
hold on
ksdensity(B_hat_x_1_sim(:,11))
hold on
line([mean(B_hat_x_1_sim(:,1)) mean(B_hat_x_1_sim(:,1))],ylim,'Color','black')
hold on
line([mean(B_hat_x_1_sim(:,11)) mean(B_hat_x_1_sim(:,11))],ylim,'Color','black')
title('The Effect of Omitting a Variable on the Distribution of the OLS estimator')
legend('No correlation','Almost perfect correlation','B_hat_sim_mean')
ylabel('Density');
xlabel('B_hat_x_1')
```

12. Additional experiments

You can explore how changing aspects of the simulation conducted above changes the nature of the results. Increase the sample size for each draw in the simulation. What do you conclude? This kind of exploration will help you see the real nature of your DGP and how your chosen statistical estimator performs.

Consider another experiment. Include a third independent variable to the regression model (`x_3`), and consider that it is not correlated with the omitted variable (`x_2`). When you inspect the sampling distribution of the coefficient estimate of `x_3`, what do you conclude?