Exercise – Understanding measurement error using simulation

1. Aim of the exercise

Measurement error in an independent variable can introduce bias into OLS coefficient estimates by creating correlation between the variable and the error term. Such errors may arise, for example, from data processing issues or respondent inaccuracies. They can undermine statistical inference. In this exercise, we simulate the effects of measurement error for the sampling distribution of the OLS estimator to examine its impact.

2. Theory

Consider the linear regression model

$$y_i = x_i^* \beta + \varepsilon_i.$$

Suppose that x_i^* is the true variable we do not observe, and what we observe is x_i, a noisy version of x_i^* with unobserved measurement error ω_i such that

$$x_i = x_i^* + \omega_i.$$

Replace x_i^* in the model so that the model includes the observable x_i and becomes estimable:

$$y_i = x_i \beta - \omega_i \beta + \varepsilon_i.$$

The last two terms are unobserved meaning that they define the new error of the regression, which we denote as ε_i^*.

In the new model x_i dependens on ε_i^* through ω_i. This makes the expected value of ε_i^* conditional on x_i nonzero. This violates the zero conditional mean assumption, and makes the OLS estimator of β biased due to measurement error.

3. Set the parameters of the simulation

To investigate the impact of measurement error on regression estimates, we begin by defining the parameters of the simulation. Clear the workspace. Specify the number of simulation replications and the sample size for each run. More replications allow us to observe how the estimated coefficients vary across different randomly generated datasets, helping us assess the consistency and reliability of the results. A larger sample size within each replication improves the accuracy of individual estimates by reducing random noise.

Next, we define the true value of the slope coefficient for the linear regression model. This known parameter serves as a benchmark against which biased estimates will be compared. To simulate the independent variable, we generate a vector of uniformly distributed random values between –1 and 1. This range ensures that the variable spans both negative and positive values evenly, providing enough variation for meaningful regression analysis. At the same time, the uniform distribution keeps the spread predictable and bounded, so the simulation remains well-behaved and avoids extreme outliers.

Finally, we introduce a vector of measurement error levels, ranging from 0 to 1 in increments of 0.1. These values allow us to systematically evaluate how increasing levels of measurement error bias the OLS estimate of the true regression coefficient.

```
%% 3. Set the parameters of the simulation

% 3.1. Clear the memory
clear;

% 3.2. Set the number of simulations
N_sim = 1000;

% 3.3. Set the sample size
N_obs = 1000;

% 3.4. Set true values for the slope
B_true = 0.5;

% 3.5. Create the systematic component of the regression
X = random("Uniform",-1,1,[N_obs 1]);

% 3.6. Level of measurement error in terms of the SD of random noise
measurement_error_level = 0:0.1:1;
```

4. Nested for loops for simulation and measurement error level

Here we conduct a Monte Carlo simulation using nested loops to examine how increasing levels of measurement error in the independent variable affect the bias of the OLS slope estimate. First, two matrices are preallocated. One stores the estimated coefficients from each simulation run, and the other stores all estimates across different levels of measurement error.

The outer loop iterates over the vector of predefined measurement error levels, ranging from 0 to 1 in increments of 0.1. Each value in this vector sets the variance of the measurement error that will be added to the independent variable. To simulate this, we generate random noise from the standard normal distribution and multiply it by the square root of the chosen measurement error level. This scaling adjusts the variance of the noise to match the specified level, since multiplying a standard normal variable by a constant increases its variance by the square of that constant. For example, if the measurement error level is set to 0.4, multiplying standard normal noise by square root of 0.4 produces noise with variance 0.4, exactly the intended distortion. The scaled noise is then added to the true independent variable to create a contaminated version that reflects the specified level of measurement error

The inner loop then performs repeated simulations. In each iteration, a new error term is generated and used to compute the true data-generating process, producing the outcome variable based on the uncontaminated regressor. However, when estimating the regression, we intentionally use a noisy version of the regressor, one that includes measurement error, while still regressing on the true outcome. This setup highlights the core problem: we are trying to explain a correctly generated outcome using a contaminated regressor, which allows us to observe how measurement error in the regressor biases the estimated coefficients. The resulting slope estimate is stored, and after all replications for a given error level are complete, the full set of estimates is saved.

This structure allows us to systematically assess how measurement error biases the OLS coefficient, quantifying attenuation as the error level increases.

```
%% 4. Nested for loops for simulation and measurement error level

% 4.1. Preallocate matrix to store OLS coefficient estimates
B_hat = NaN(N_sim,1);

% 4.2. Preallocate matrix to store estimates across mea. err. levels
B_hat_measurement_error_level = NaN(N_sim, ...
    1,length(measurement_error_level));

% 4.3. Nested for loops for simulation and measurement error level
for j = 1:length(measurement_error_level)
    X_with_measurement_error = X+random('Normal',0,1,[N_obs 1]) ...
        *sqrt(measurement_error_level(j));
    for i = 1:N_sim
        u = random('Normal',0,1,[N_obs 1]);
        y = X*B_true+u;
        LSS = exercisefunctionlss(y,X_with_measurement_error);
        B_hat(i,1) = LSS.B_hat(1,1);
    end
    B_hat_measurement_error_level(:,:,j) = B_hat;
end
```

5. Plot the sampling distribution of the OLS estimator

This section visualizes how different levels of measurement error affect the sampling distribution of the estimated slope coefficient. Using kernel density estimation, we plot the distributions across repeated simulations for three selected error levels: 0 representing no error, 0.5 representing moderate error, and 1 representing maximum error in this setup. These error levels control the variance of the noise added to the independent variable. In each case, the regression is estimated using the contaminated regressor while keeping the outcome variable fixed from the true data-generating process. The figure shows that when there is no measurement error, the estimates are tightly centered around the true value of 0.5, indicating no bias. As the error level increases, the slope estimate becomes biased toward zero. This is classic attenuation bias. The more noise in the regressor, the weaker the apparent relationship with the outcome, so the estimated slope shrinks. As the error level increases, the distributions become less dispersed, not because they become accurate, but because the regressor is so noisy that the model has no choice but to repeatedly guess near zero.

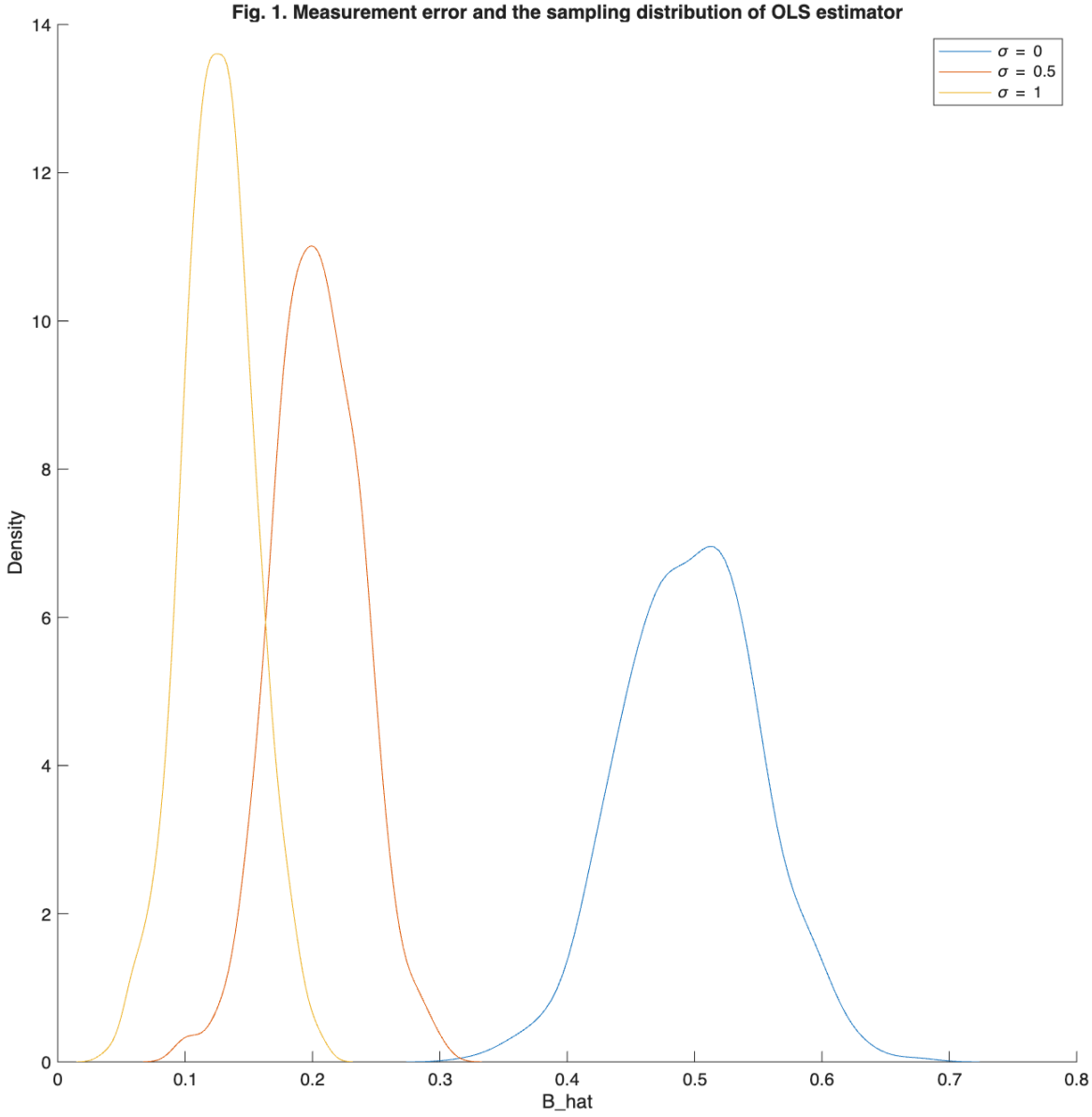
```
%% 5. Plot the sampling distribution of the OLS estimator

% 5.1. Kernel density estimation
[f1,x1] = ksdensity(B_hat_measurement_error_level(:,1,1));
[f2,x2] = ksdensity(B_hat_measurement_error_level(:,1,6));
[f3,x3] = ksdensity(B_hat_measurement_error_level(:,1,11));

% 5.2. Sampling distribution of OLS estimator subject to mea. error
figure
hold on
plot(x1,f1,'DisplayName','Measurement error variance = 0');
```

```
plot(x2,f2,'DisplayName','Measurement error variance = 0.5');
plot(x3,f3,'DisplayName','Measurement error variance = 1');
xlabel('B\_hat');
ylabel('Density');
title(['Fig. 1. Effect of measurement error on ' ...
    'the sampling distribution of OLS estimator']);
legend('show');
hold off
```

Fig. 1. Measurement error and the sampling distribution of OLS estimator



15. Final notes