Exercise – Understanding heteroskedasticity using simulation

1. Aim of the exercise

In linear regression analysis it is assumed that the errors are spherical. This exercise uses simulation to study the implications of violating this assumption for the sampling distribution of the OLS estimator.

2. Theory

In linear regression analysis, the homoskedasticity assumption requires that the each error term of the regression, ε_i, has the same finite variance, σ^2, at given values of an explanatory variable. That is:

$$\mathrm{Var}\left[\varepsilon_i \mid x_i\right] = \sigma^2,\ \forall\, i.$$

Non-constant error variance means heteroskedasticity. Non-constant error variance is an efficiency problem because the model does not predict the dependent variable reliably at certain values of an explanatory variable.

3. Application

3.1. Clear the memory

Clear the memory from possible calculations from an earlier session.

```
% 3.1. Clear the memory
clear;
```

3.1. Set the number of simulations

Set the number of simulations to be carried out.

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

3.3. 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.

```
% 3.3. Set the sample size
N_obs = 500;
```

3.4. Set true values for the coefficients of the model

Assume that we know the true values of the coefficients of the model.

```
% 3.4. Set true values for the coefficients
B_true = [0.2 0.5]';
```

3.5. Generate data for the independent variable

Create the constant term. Draw random numbers from the uniform distribution, and require them to be in the range $[-1, 1]$. Consider this vector as the independent variable of the regression model. Create the systematic component of the regression equation, and call it `X`.

```
% 3.5. Create the systematic component of the regression
x_0 = ones(N_obs,1);
x_1 = random('Uniform',-1,1,[N_obs 1]);
X = [x_0 x_1];
```

3.6. Define the number of coffcients to be simulated.

Define the number of coeffcients to be simulated.

```
% 3.6. Define the number of parameters to be estimated
N_par = 2;
```

3.7. Preallocate matrices for storing simulated statistics

Preallocate matrices that will store the simulated coefficient estimates generated under heteroskedasticity and homoskedasticity. Each matrix is `N_par` $\times$ `N_sim` because we have `N_par` coefficients to estimate, and `N_sim` coefficients to simulate. Preallocate also a vector that will store in each row a simulated standard deviation of the residuals from a model with heteroskedastic errors. The reason of creating this vector will be explained in a later section.

```
% 3.7. Preallocate matrices for storing simulated statistics
B_hat_sim_het = NaN(N_sim,N_par);
B_hat_sim_hom = NaN(N_sim,N_par);
sigma_hat_sim_het = NaN(N_sim,1);
```

3.8. Heteroskedasticity parameter

To produce heteroskedasticity, we need to simulate an error for the data generating process (DGP) that does not have a constant variance across the observations of an explanatory variable. In the exercise on the sampling distribution of the OLS estimator, we used a value of `1` as the standard deviation of the error term. Here we replace it with `exp(x_1*Gamma)`. We use the exponential distribution because the exponential of any number will always be positive so that we do not generate negative variance. Our independent variable of interest is (`x_1`) multiplied by `Gamma`. We set `Gamma` to `1.5`. This is an arbitrary choice. We can explore the impact of changing `Gamma`. This setup renders the error variance a function of `x_1`. In particular, larger values of `x_1` will be associated with a larger variance in the error of the DGP compared to smaller values of `x_1`.

```
% 3.8. Heteroskedasticity parameter
Gamma = 1.5;
```

3.9. Sampling distribution of the OLS estimator under heteroskedasticity

Here we calculate coefficient estimates from repeated samples generated by the assumed DGP. The error term of the DGP is heteroskedastic. We also calculate the standard deviation of the residuals in each repeated sample in the simulation. The coefficient estimates and the standard deviation of the residuals are calculated using the function `exercisefunctionlss`.

```
% 3.9. Sampling distribution of the OLS estimator when errors are het.
for i = 1:N_sim
    u_het = random('Normal',0,exp(x_1*Gamma),[N_obs 1]);
    y_het = X*B_true+u_het;
    LSS_het = exercisefunctionlss(y_het,X);
    B_hat_sim_het(i,1) = LSS_het.B_hat(1,1);
    B_hat_sim_het(i,2) = LSS_het.B_hat(2,1);
    sigma_hat_sim_het(i,1) = LSS_het.sigma_hat;
end
```

3.10. Average of standard deviation estimate generated under heteroskedasticity

Average of standard deviation estimate generated under heteroskedasticity.

```
% 3.10. Average of standard deviation estimate under heteroskedasticity
sigma_hat_sim_het_mean = mean(sigma_hat_sim_het);
```

3.11. Sampling distribution of the OLS estimator under homoskedasticity

A valid comparison of the simulations based on heteroskedastic and homoskedastic errors requires care. In the preceding simulation, we saved the estimate of sigma, `LSS_het.sig_hat`, from repeated samples in the vector array `sig_sim_het`. The mean of this estimate is about 1.8. In the current simulation under homoskedasticity, if we set the standard deviation of the error term to 1, not surprisingly, it will produce an average value over 1,000 repetitions very close to 1. Therefore, if we simply compare the simulation output under homoskedasticity and heteroskedasticity, two parameters will actually be changing: first the overall variance of the error term, and second heteroskedasticity. Therefore, if we find differences between the two simulations, we may not be able to study whether they emerge due to heteroskedasticity or just from the difference in the average size of `sigma`. Our aim is to make a comparison where only heteroskedasticity is changing.

In the simulation based on heteroskedastic erors, we named the array for the dependent variable as `y_het`. We then estimated an OLS regression. The output of the function is stored in the array `LSS_het`. Next, we stored the coefficient estimates from the model of the homoskedastic errors in the array `B_hat_sim_het`. Here, using homoskedastic errors, we store the dependent variable in array `y_hom`. Next we estimate an OLS regression. We store the output of this function in array `LSS_hom`. Next, we store the coefficient estimates from the model of homoskedastic errors in `B_hat_sim_hom`. However, when creating the array `y_hom`, or `u_hom`, we use as the standard deviation of the error term `sigma_sim_het_mean` which is the average of that based on the simulation using heteroskedastic errors which is about 1.8. This ensures that the overall variance of the error term does not change, on average, between the two simulations with and without heteroskedasticity. The only difference between them is that one, that is `y_het`, includes heteroskedasticity, and the other, that is `y_hom`, does not.

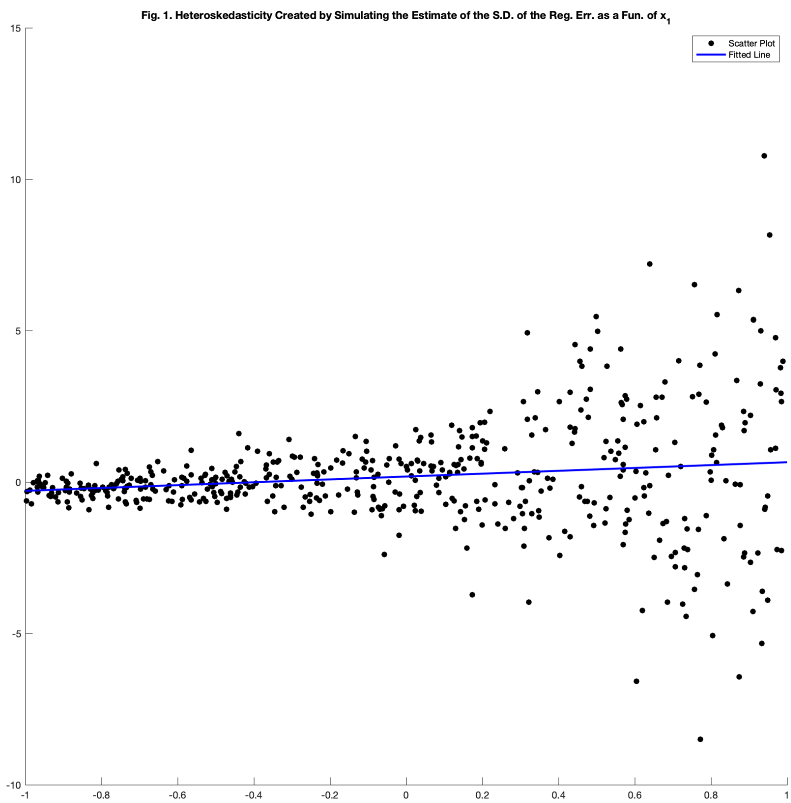
```
% 3.11. Sampling distribution of the OLS estimator when errors are hom
```

```
for i = 1:N_sim
    u_hom = random('Normal',0,sigma_hat_sim_het_mean,[N_obs 1]);
    y_hom = X*B_true+u_hom;
    LSS_hom = exercisefunctionlss(y_hom,X);
    B_hat_sim_hom(i,1) = LSS_hom.B_hat(1,1);
    B_hat_sim_hom(i,2) = LSS_hom.B_hat(2,1);
end
```

4. Plot the scatter diagram and the OLS fitted line

Here we plot y against x_1 and the OLS regression line. Observe that the spread of the points increases as x_1 increases.

```
%% 4. Plot the scatter diagram and the OLS fitted line
figure
hold on
scatter(X(:,2),y_het,'filled','black')
set(lsline,'color','blue','LineWidth',2)
title(['Fig. 1. Heteroskedasticity Created by Simulating the ' ...
    'Estimate of the S.D. of the Reg. Err. as a Fun. of x_1'])
legend('Scatter Plot','Fitted Line');
hold off
```

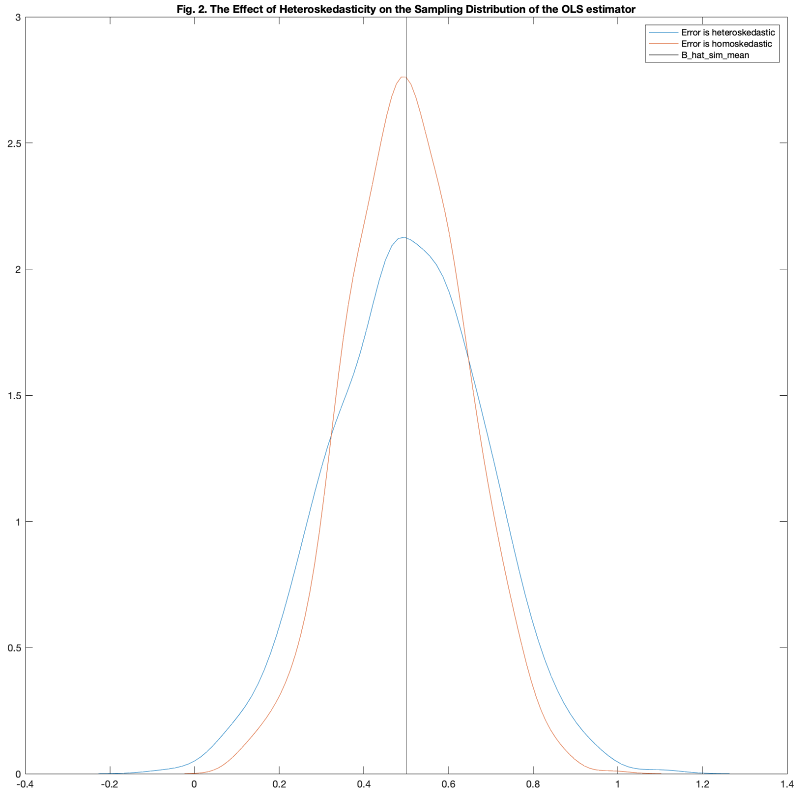


5. Plot the sampling distribution of the OLS estimator

Here we compare the estimates from the two models with homoskedastic and heteroskedastic errors. Notice that the density of estimates both with and without heteroskedasticity show unbiasedness. That is, the peaks of the distributions are centered at the true parameter values. However, there is a noticeable difference in the spread of the distributions. The estimates generated under heteroskedasticity have less density concentrated near the true value and more density farther away. This is graphical evidence of the efficiency problem that heteroskedasticity generates. When the variance of the error term is a function of an independent variable, that is if it is not constant, any single estimate of a coefficient on that independent variable is less likely to be close to the true parameter compared to when the error variance is constant. This phenomenon does not extend to the intercept term because it does not operate on any independent variable.

```
%% 5. Plot the sampling distribution of the OLS estimator
figure
hold on
ksdensity(B_hat_sim_het(:,2))
ksdensity(B_hat_sim_hom(:,2))
line([mean(B_hat_sim_hom(:,2)) mean(B_hat_sim_hom(:,2))],ylim, ...
    'Color','black')
title(['Fig. 2. The Effect of Heteroskedasticity on the Sampling ' ...
    'Distribution of the OLS estimator'])
legend('Error is heteroskedastic','Error is homoskedastic', ...
    'B\_hat\_sim\_mean');
hold off
```

Fig. 2. The Effect of Heteroskedasticity on the Sampling Distribution of the OLS estimator



6. Check the variance of the heteroskedastic error

The code below illustrates how the variance of the heteroskedastic error increases with the increasing values of the regressor.

```
var(u_het(x_1 > 0.1 & x_1 < 0.3,1))
var(u_het(x_1 > 0.6 & x_1 < 0.8,1))
```

7. Final notes

This file is prepared and copyrighted by Tunga Kantarcı. Parts of the simulation exercise are based on Carsey, T. M., and Harden, J. J., 2014. Monte Carlo simulation and resampling methods for social science. SAGE Publications. This file and the accompanying MATLAB files are available on GitHub and can be accessed via this link.