Empirical exercise – Violation of the spherical errors assumption with heteroskedasticity

1. Aim of the exercise

To understand the implications of violating the spherical errors assumption with heteroskedasticity 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 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;
```

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

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

6. Generate data for the independent variable

Create the constant term. Draw a set of random numbers from the uniform distribution, and require the numbers 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`.

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

7. Create empty matrices and a vector array to store the simulated OLS coefficient estimates and the simulated sigma

Create an empty matrix that will store the simulated coefficient estimates generated under heteroskedasticity, and another one for the coefficient estimates generated under homoskedasticity. Each matrix is `N_par` $\times$ `N_sim` because we have `N_par` coefficients to estimate, and `N_sim` coefficients to simulate. Create also an empty 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.

```
N_par = 2;
B_hat_sim_hete = NaN(N_sim,N_par);
B_hat_sim_homo = NaN(N_sim,N_par);
sigma_hat_sim_hete = NaN(N_sim,1);
```

8. Heteroskedasticity parameter

OLS assumes that the variance of the dependent variable conditional on the model (independent variable and the coefficients) is constant. Stated more simply, OLS assumes that the variance of the residuals is constant. The term for constant variance is homoskedasticity. In contrast, heteroskedasticity means non-constant variance. Hence, the presence of heteroskedasticity in the residuals of an OLS regression constitutes a violation of an OLS assumption. More specifically, heteroskedasticity presents problems for OLS when the variance of the residuals is a function of one or more of the independent variables. Non-constant error variance is an efficiency problem because the model does not predict the dependent variable as reliably at certain values of the independent variables. We can use simulation to illustrate this more clearly.

To produce heteroskedasticity, we need to simulate a residual for the data generating process (DGP) that does not have a constant variance. In particular, we want to simulate a DGP where residual variance is a function of one or more of the independent variables. In the previous exercise on the sampling distribution of the OLS estimator, we have 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, which is helpful because there is no such thing as a negative variance. The other part of the formula is our independent variable of interest (`x_1`) multiplied by some parameter (in this case with `Gamma`). In this example, we set the parameter `Gamma` to `1.5`. This is an arbitrary choice: you can explore the impact of changing the value of the `Gamma` on the results produced by the simulation. This setup renders the error variance a function of `x_1`. In particular, larger values of `x_1` will be associated with larger variance in the error term of the DGP compared to smaller values of `x_1`.

```
Gamma = 1.5;
```

9. Create the sampling distribution of the OLS estimator under heteroskedasticity

The code at the end of the section calculates coefficient estimates for the intercept term and the independent variable from repeated samples generated by the assumed DGP. The error term of the DGP is heteroskedastic. The code also calculates the standard deviation of the residuals from each sample of the repeated samples in the simulation. The coefficient estimates and the standard deviation of the residuals are calculated using the supplied function `exercisefunctionlss`. Inspect the content of this function briefly to convince yourself about the calculations of the parameters thereof.

```
for i = 1:N_sim
    u_hete = normrnd(0,exp(x_1*Gamma),N_obs,1);
    y_hete = X*B_true+u_hete;
    LSS_hete = exercisefunctionlss(y_hete,X);
    B_hat_sim_hete(i,1) = LSS_hete.B_hat(1,1);
    B_hat_sim_hete(i,2) = LSS_hete.B_hat(2,1);
    sigma_hat_sim_hete(i,1) = LSS_hete.sigma_hat;
end
```

10. Plot the scatter diagram and the OLS fitted line

Plot `y` against `x_1` with the OLS regression running through the points. Notice that the spread of the points increases dramatically as `x_1` increases. This illustrates the change we have made using one sample of the simulated data.

```
scatter(X(:,2),y_hete,'filled','black')
hold on
set(lsline,'color','blue','LineWidth',2)
hold off
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');
```

11. Create the sampling distribution of the OLS estimator under homoskedasticity

Now that we have seen the basic idea behind changing a DGP in a simulation, we can examine the consequences of this change for the coefficient estimates. We do this by comparing the distributions of the coefficient estimates obtained above with those from a basic OLS simulation with no assumptions violated: that is, the errors are homoskedastic.

When carrying out this basic OLS simulation, we need to make an adjustment in this simulation to make a 'fair' comparison. Notice from the simulation above that we saved the estimate of sigma (`LSS_hete.sig_hat`) from repeated samples in the vector array `sig_sim_hete`. As it turns out, the mean in that simulation is about 1.8. In the basic OLS simulation (under

homoskedasticity) we are to carry out in this section, if we set the standard deviation of the error term to 1, it will, not surprisingly, produce an average value over 1,000 repetitions very close to 1. Hence, if we simply compare the basic OLS simulation with the heteroskedasticity simulation, two parameters will actually be changing: (l) the overall variance of the error term, and (2) heteroskedasticity. Hence, if we see differences between the two simulations, we may not be able to say whether they emerge due to heteroskedasticity or just from the difference in the average size of `sigma`. We want to make a comparison where only heteroskedasticity is changing.

Above, using the heteroskedastic DGP, we have created the dependent variable `y_hete`. We estimated an OLS regression using the function `exercisefunctionlss`. The output of the function is stored in `LSS_hete`. Then, we stored the coefficient estimates from the model of the homoskedastic DGP in `B_hat_sim_hete`. Here, using the homoskedastic DGP, we create the dependent variable `y_homo`. We then estimate an OLS regression using the function `exercisefunctionlss`. We store the output of the function in `LSS_homo`. Then, we store the coefficient estimates from the model of the homoskedastic DGP in `B_hat_sim_homo`. However, when creating `y_homo`, or more specifically when creating `u_homo`, we use as the standard deviation of the error term `sigma_sim_hete_mean` which is the average from the heteroskedastic simulation (about 1.8). This ensures that the overall variance of the error term does not change, on average, between the two DGPs (with and without heteroskedasticity). The only difference between them is that one includes heteroskedasticity (`y_hete`) and one does not (`y_homo`).

```
sigma_hat_sim_hete_mean = mean(sigma_hat_sim_hete);
for i = 1:N_sim
    u_homo = normrnd(0,sigma_hat_sim_hete_mean,N_obs,1);
    y_homo = X*B_true+u_homo;
    LSS_homo = exercisefunctionlss(y_homo,X);
    B_hat_sim_homo(i,1) = LSS_homo.B_hat(1,1);
    B_hat_sim_homo(i,2) = LSS_homo.B_hat(2,1);
end
```

12. Plot the sampling distribution of the OLS estimator

Now, we can compare the estimates from the two DGPs. To do this, we do not use histograms but kernel density estimates. A kernel density is a smoothed histogram represented by a line. We do this to allow for better visibility when comparing two distributions. The graph below plots the density of coefficient estimates for the coefficient estimate of the independent variable both with and without heteroskedasticity.

Notice that the density of estimates both with and without heteroskedasticity show unbiasedness: 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 creates. When the variance of the error term is a function of an independent variable (that is, 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 with when the error variance is constant. This phenomenon does not extend to the intercept term because it does not operate on any inde-

pendent variable.

```
ksdensity(B_hat_sim_hete(:,2))
hold on
ksdensity(B_hat_sim_homo(:,2))
hold on
line([mean(B_hat_sim_homo(:,2)) mean(B_hat_sim_homo(:,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');
```