Exercise – Understanding the normality assumption of the regression error

1. Aim of the exercise

In finite samples, the errors of the linear regression model are assumed to follow a normal distribution. This exercise examines the implications for the sampling distribution of the OLS estimator when the regression errors do not follow a normal distribution.

2. Application

2.1. Clear the memory

Clear the memory from possible calculations from an earlier session.

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

2.2. Set the number of simulations

Set the number of simulations to be carried out.

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

2.3. 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.

```
% 2.3. Set the sample size
N_obs = 350;
```

2.4. Set true values for the coefficients

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.

```
% 2.4. Set true values for the coefficients
B_true = [0.2 3.5]';
```

2.5. Define the number of coefficients to be simulated

Define the number of coefficients to be simulated.

```
% 2.5. Define the number of coefficients to be simulated
N_par = 1;
```

2.6. Create the systematic component of the regression equation

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`.

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

2.7. Preallocate matrices for storing statistics to be simulated

Preallocate matrices that will store simulated coefficient estimates generated under different distributional assumptions for the error, using different estimators. Each matrix is `N_sim` $\times$ `N_par` because we simulate `N_sim` times the coefficient of the independent variable `x_1`, and we have `N_par` coefficients to simulate.

```
B_hat_1_sim_OLS_normal = NaN(N_sim,N_par);
B_hat_1_sim_IRLS_normal = NaN(N_sim,N_par);
B_hat_1_sim_OLS_t = NaN(N_sim,N_par);
B_hat_1_sim_IRLS_t = NaN(N_sim,N_par);
```

2.8. Degrees of freedom of the t distribution

The t distribution depends the degrees of freedom parameter. The parameter controls the kurtosis of the distribution. Create this parameter, and assume an experimental value of 2 for it. We will use this parameter to generate random draws from the t distribution later in our simulation.

```
% 2.8. Define the degrees of freedom for t distribution
t_df = 2;
```

2.9. Create sampling distributions for the OLS and IRLS estimators based on errors with different distributions

We will examine the implications of violating the normality assumption. There are many ways in which this assumption can be violated, but here we focus on only one. In particular, we study the consequences for the sampling distribution of the OLS estimator when the true error distribution has heavier tails than the normal distribution. Distributions with heavy tails are important to social scientists because they are more likely to generate observations that are outliers compared with what one would expect from the normal distribution. In this example, we compare the performance of the OLS estimator with that of an alternative estimator that is robust to outliers. One distribution that has heavier tails than the standard normal distribution is the t distribution.

How does assuming that the error term follows a t distribution effect estimation? To answer this question we will compare the performance of the OLS estimator with that of an alternative estimator. This alternative estimator is the Iteratively Reweighted Least Squares estimator (IRLS), which produces robust estimates when outlying observations are present: https://

en.wikipedia.org/wiki/Iteratively_reweighted_least_squares. By minimizing the sum of absolute residuals rather than squared residuals like OLS, IRLS is not disproportionately influenced by outliers. Indeed, while outliers can have a substantial effect on OLS estimates, their impact on those of IRLS is smaller. As a result, IRLS is a more efficient estimator than the OLS estimator under this circumstance. IRLS allows the analyst to handle heavy-tailed data without giving special treatment to outliers or just deleting them.

To illustrate this, we simulate a model with errors drawn from standard normal distribution, and one with errors drawn from the t distribution. The t distribution has one parameter: degrees of freedom. Above, we have set it to 2. After creating a DGP with standard normal errors and a DGP with t errors, we estimate models on both types of data using the OLS and IRLS estimators. For this, we use the built-in `robustfit` function of MATLAB that offers to produce standard OLS estimates as well as IRLS estimates if the option for the estimation method is specified accordingly in the function syntax of the `robustfit` function.

Consider the for loop presented at the end of the section. We specify a DGP with errors that have a standard normal distribution, and another with errors that have a t distribution. We then estimate the regression with standard normal errors, using the OLS estimator. Next, we estimate the same regression using the IRLS estimator. Do you expect these estimators to produce similar coefficient estimates? In the remaining lines, we estimate the regression with errors that have a t distribution, using the OLS and IRLS estimators. In this case, do you expect these estimators to produce similar coefficient estimates?

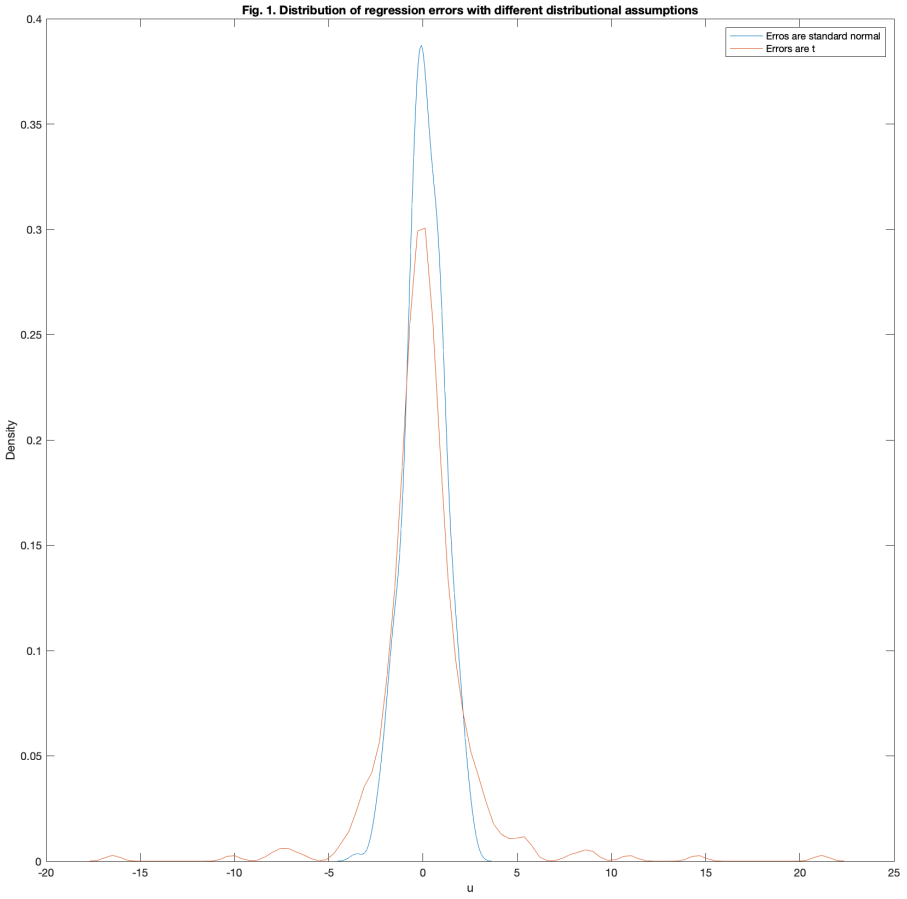
```
% 2.9. Create sampling distributions for the OLS and IRLS estimators
for i = 1:N_sim
    u_normal = random('Normal',0,1,[N_obs 1]);
    y_normal = X*B_true+u_normal;
    u_t = random('t',t_df,[N_obs 1]);
    y_t = X*B_true+u_t;
    OLS = robustfit(x_1,y_normal,'ols');
    B_hat_1_sim_OLS_normal(i,1) = OLS(2,1);
    IRLS = robustfit(x_1,y_normal,'bisquare');
    B_hat_1_sim_IRLS_normal(i,1) = IRLS(2,1);
    OLS = robustfit(x_1,y_t,'ols');
    B_hat_1_sim_OLS_t(i,1) = OLS(2,1);
    IRLS = robustfit(x_1,y_t,'bisquare');
    B_hat_1_sim_IRLS_t(i,1) = IRLS(2,1);
end
```

3. Plot example distributions of errors with different distributional assumptions

In the previous section, in the for loop, we have generated two types of errors. One with a standard normal distribution, and another one with a t distribution. Plotted here are kernel smoothed histograms of these errors. The distributions result from the last draw in the for loop. Looking at these two distributions, what do you conclude?

```
%% 3. Plot errors with different distributional assumptions
ksdensity(u_normal)
hold on
ksdensity(u_t)
title(['Fig. 1. Distribution of regression errors with different ' ...
```

```
    'distributional assumptions'])
legend('Erros are standard normal','Errors are t')
ylabel('Density')
xlabel('u')
hold off
```

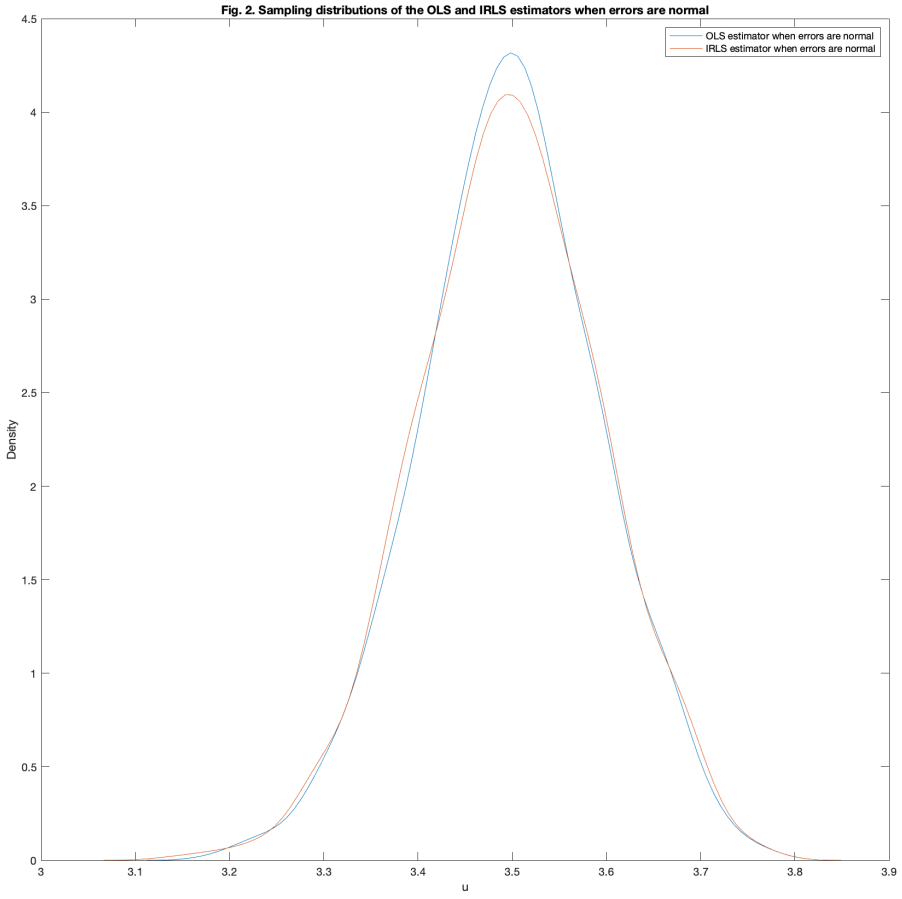


4. Plot the sampling distributions of the OLS and IRLS estimators when errors are normal

Plot the sampling distributions of the OLS and IRLS estimators when the errors of the regression are assumed to be standard normal. The two distributions are close to each other. Is this a surprising result?

```
%% 4. Plot the sampling distributions of estimators when errors are normal
ksdensity(B_hat_1_sim_OLS_normal(:,1))
hold on
ksdensity(B_hat_1_sim_IRLS_normal(:,1))
title(['Fig. 2. Sampling distributions of the OLS and IRLS ' ...
    'estimators when errors are normal'])
legend('OLS estimator when errors are normal',['IRLS estimator ' ...
```

```
    'when errors are normal'])
ylabel('Density')
xlabel('u')
hold off
```



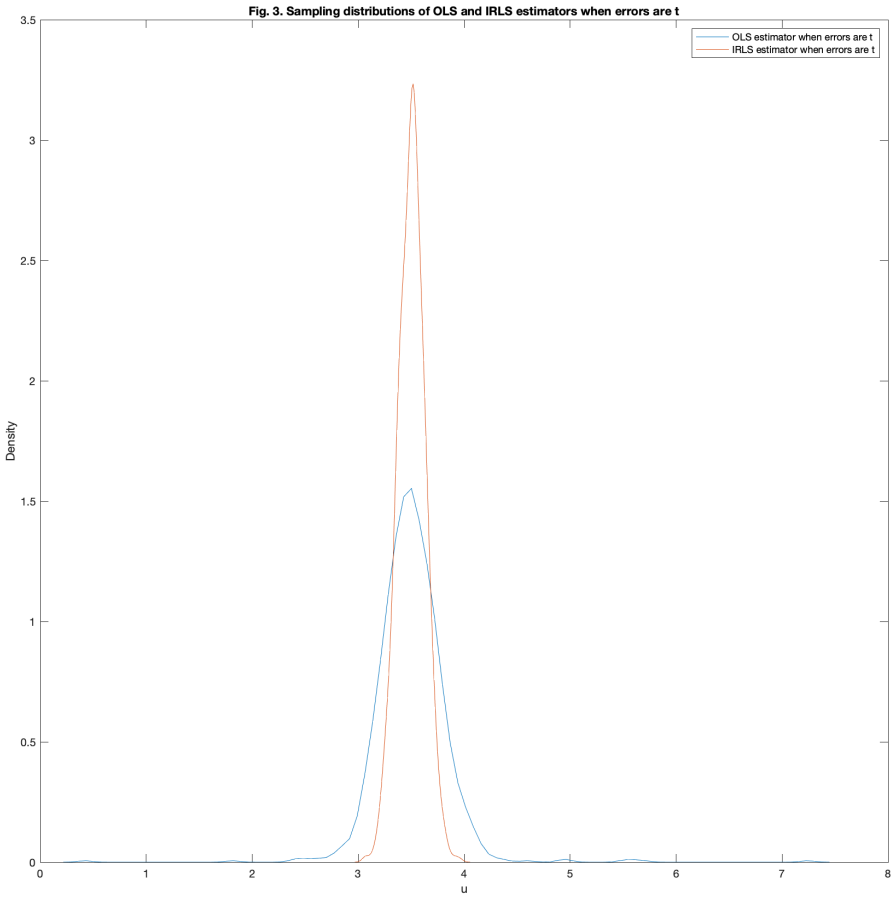
5. Plot the sampling distributions of the OLS and IRLS estimators when errors are t

Here we plot the sampling distributions of the OLS and IRLS estimates when the errors of the regression follow a t distribution. Given the samples size of `N_obs` observations, the OLS estimator appears to be less efficient.

Remember that the OLS estimator is BLUE (when the variance-covariance matrix has a scalar form). That is, apart from being an unbiased estimator, it is the most efficient estimator. But the plot suggests that it is not more efficient than the IRLS estimator when the errors are t distributed. So how can we still claim that the OLS estimator is the most efficient? We need to realize that the OLS estimator is a linear estimator while the IRLS estimator is a nonlinear estimator. So we cannot compare the two estimators based on their efficiency because they are not both linear or non-linear estimators.

```
%% 5. Plot the sampling distributions of estimators when errors are t
```

```
ksdensity(B_hat_1_sim_OLS_t(:,1))
hold on
ksdensity(B_hat_1_sim_IRLS_t(:,1))
title(['Fig. 3. Sampling distributions of OLS and IRLS estimators ' ...
    'when errors are t'])
legend('OLS estimator when errors are t',['IRLS estimator ' ...
    'when errors are t'])
ylabel('Density')
xlabel('u')
hold off
```



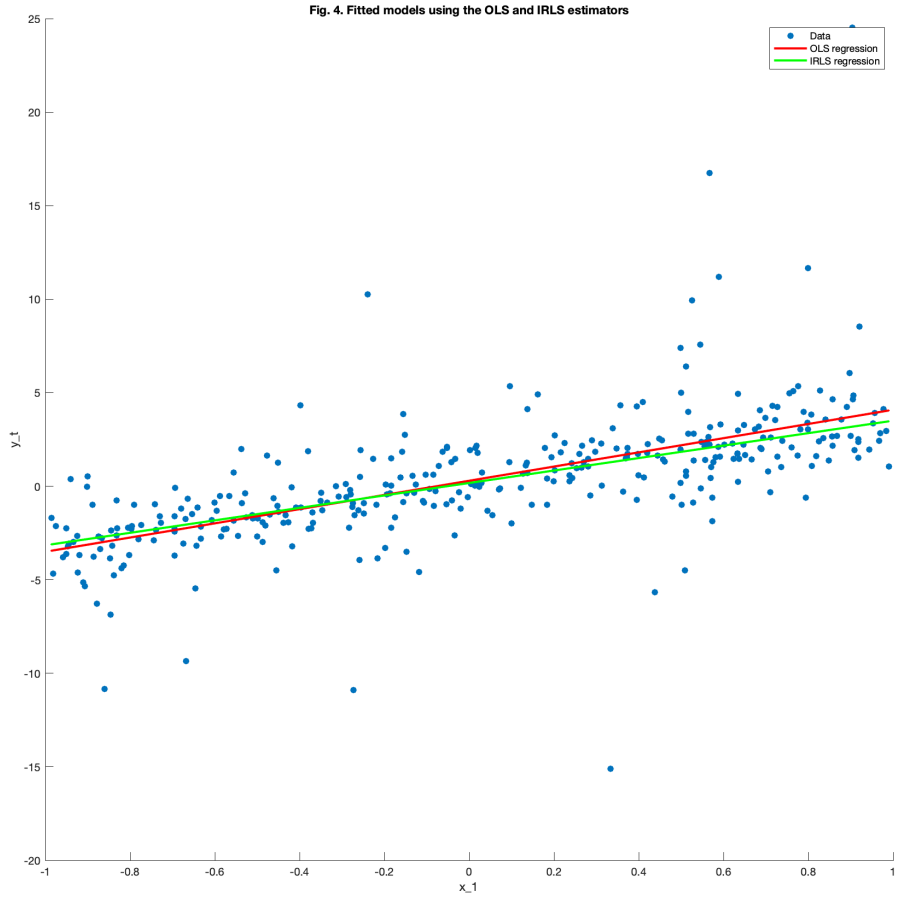
6. Plot the scatter plot and two regression lines fitted using the OLS and IRLS estimators

Here we overlay two types of plots. The first is a scatter plot of the dependent variable against the independent variable. The dependent variable is that of the DGP that imposes a t distribution on the errors. This scatter plot is overlaid by two regression lines fitted using the OLS and the IRLS estimators. The fitted line using the OLS estimator appears to be influenced by the outliers. On the other hand, the fitted line using the IRLS estimator is robust to the outliers.

Increase the sample size `N_obs` and execute the simulation again. The plot will show a change. What would explain this change?

Increase the degrees of freedom of the t distribution from 2 to, for example, 5. The plot will show a change. What would explain this change?

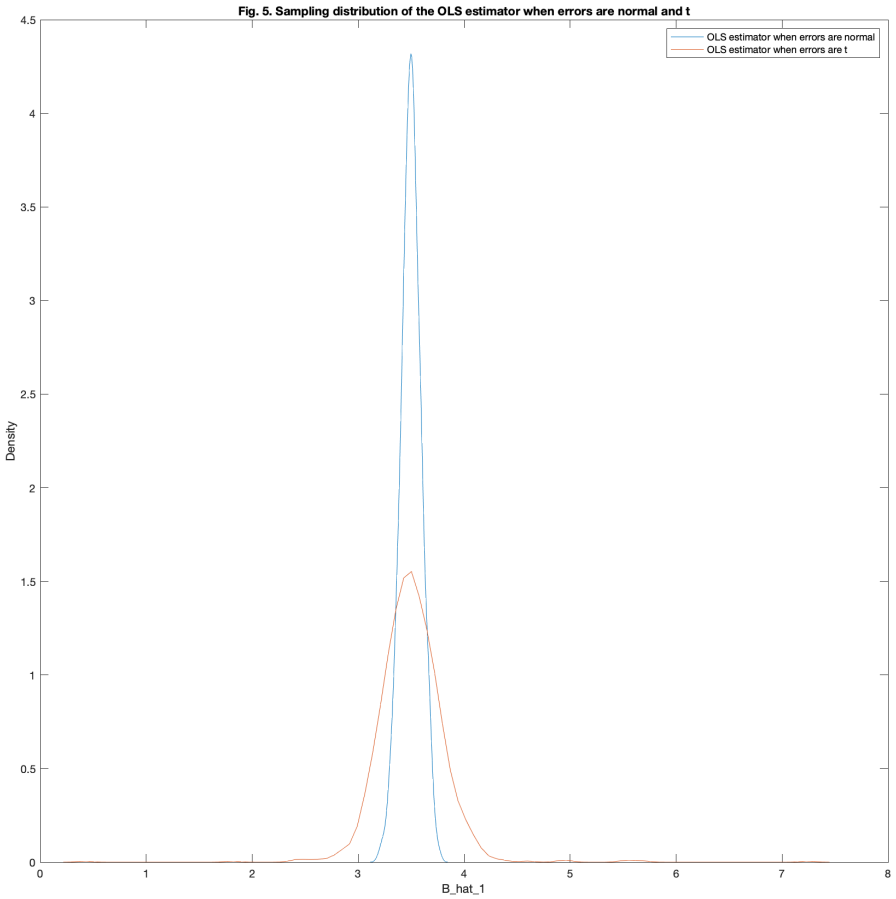
```
%% 6. Plot the fitted models using the OLS and IRLS estimators
scatter(x_1,y_t,'filled');
grid on;
hold on
y_t_hat_OLS = OLS(1)+OLS(2)*x_1;
y_t_hat_IRLS = IRLS(1)+IRLS(2)*x_1;
plot(x_1,y_t_hat_OLS,'red','LineWidth',2);
plot(x_1,y_t_hat_IRLS,'green','LineWidth',2)
title('Fig. 4. Fitted models using the OLS and IRLS estimators')
legend('Data','OLS regression','IRLS regression')
grid off;
ylabel('y\_t')
xlabel('x\_1')
hold off
```



7. Other simulation experiments

Plot the sampling distribution of the OLS estimator when regression errors are normal and when they are t. What do you conclude?

```
%% 7. Other simulation experiments
ksdensity(B_hat_1_sim_OLS_normal(:,1))
hold on
ksdensity(B_hat_1_sim_OLS_t(:,1))
title(['Fig. 5. Sampling distribution of the OLS estimator ' ...
    'when errors are normal and t'])
legend('OLS estimator when errors are normal',['OLS estimator ' ...
    'when errors are t'])
ylabel('Density')
xlabel('B\_hat\_1')
hold off
```



8. Final notes