Empirical exercise – Violation of the normality assumption

1. Aim of the exercise

To understand how the OLS estimator behaves when the errors of the regression are not normal.

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 = 75;
```

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; 3.5];
N_par = 1;
```

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`.

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

7. Create empty matrices for storing simulated coefficient estimates

Create empty matrices that will store simulated coefficient estimates generated under different distributional assumptions for the error, using different estimators. Each matrix is `N_sim` × `N_par` because we simulate the coefficient of the `x_1` variable `N_sim` times, and we have `N_par` coefficient to simulate. The purpose of these matrices will become more clear in a later section.

```
B_hat_sim_x_1_OLS_normal = NaN(N_sim,N_par);
B_hat_sim_x_1_IRLS_normal = NaN(N_sim,N_par);
B_hat_sim_x_1_OLS_t = NaN(N_sim,N_par);
B_hat_sim_x_1_IRLS_t = NaN(N_sim,N_par);
```

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 later in this exercise to generate random draws from the t distribution in our simulation.

```
t_df = 2;
```

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

In this exercise we examine the consequences of violating the normality assumption. There are many ways in which this assumption could be violated, but we focus on only one here.

In particular, we study the consequences for 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: `https://en.wikipedia.org/wiki/Student%27s_t-distribution`.

How does assuming that the error term follows a t distribution affect 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. We set this parameter to 2. After creating a DGP with standard normal errors and a DGP with t errors, we then estimate models on both types of data using 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. The first two lines specify a DGP with errors that have a standard normal distribution. The next two lines specify a DGP with errors that have a t distribution. In lines six and seven, we estimate the regression with standard normal errors, using the OLS estimator. In lines eight and nine, we estimate the same regression using the IRLS estimator. Do you expect these estimators to produce similar coefficient estimates? In the remaining four 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?

```
for i = 1:N_sim
    u_normal = normrnd(0,1,N_obs,1);
    y_normal = X*B_true+u_normal;
    u_t = trnd(t_df,N_obs,1);
    y_t = X*B_true+u_t;
    OLS = robustfit(x_1,y_normal,'ols');
    B_hat_sim_x_1_OLS_normal(i,1) = OLS(2,1);
    IRLS = robustfit(x_1,y_normal,'bisquare');
    B_hat_sim_x_1_IRLS_normal(i,1) = IRLS(2,1);
    OLS = robustfit(x_1,y_t,'ols');
    B_hat_sim_x_1_OLS_t(i,1) = OLS(2,1);
    IRLS = robustfit(x_1,y_t,'bisquare');
    B_hat_sim_x_1_IRLS_t(i,1) = IRLS(2,1);
end
```

10. 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. Presented at the end of section 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?

```
ksdensity(u_normal)
hold on
ksdensity(u_t)
legend('Standard normal errors','t errors')
hold off
```

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

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

```
ksdensity(B_hat_sim_x_1_OLS_normal(:,1))
hold on
ksdensity(B_hat_sim_x_1_IRLS_normal(:,1))
legend('OLS','IRLS')
hold off
```

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

The code presented at the end of the section plots the sampling distributions of the OLS and IRLS estimates when the errors of the regression are assumed to follow a t distribution. Given the samples size `N_obs`, the OLS estimator appears to be less efficient. But we know that the OLS estimator is BLUE. That is, apart from being a linear and unbiased estimator, it is the 'best' estimator, meaning that it is the most efficient estimator. But the plot suggests that it is not the best. Is something wrong?

```
ksdensity(B_hat_sim_x_1_OLS_t(:,1))
hold on
ksdensity(B_hat_sim_x_1_IRLS_t(:,1))
legend('OLS','IRLS')
hold off
```

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

The code presented at the end of the section overlays two types of graphs. The first type 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 lying beneath the fitted line. On the other hand, the fitted line using the IRLS estimator is robust to these outliers.

Increase the sample size `N_obs` to 100 and carry out the simulation above once again. The plot shows a change. What would explain this change?

Increase the degrees of freedom of the t distribution from 2 to say 5. The plot shows a change. What would explain this change?

```
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)
legend('Data','OLS Regression','IRLS Regression')
```