Empirical exercise – Inference – Understanding the t test using the simulated t distribution

1. Aim of the exercise

Hypothesis testing is typically taught using the tabulated distribution of the concerning test statistic presented at the back of a statistics textbook. This means that it is left to the student to imagine that there is in fact a distribution used when carrying out the hypothesis test. This makes the procedure of hypothesis testing unclear. This exercise simulates the distribution of the test statistic of interest so that the student can visualise the distribution itself right in front of him when conducting the hypothesis test.

The empirical context is as follows. Mincer (1974) investigates how innate ability affects wages. He estimates a regression model where the dependent variable is the logarithm of wage, and the control variables are IQ score as a proxy for innate ability, experience, education, and a quadratic function of age. The aim is to conduct a hypothesis test on the population coefficient of IQ. The data is the wage2 dataset from Wooldridge (2015).

2. Prepare the data and obtain the coefficient estimate of interest

2.1. Load the data

```
clear;
load 'M:\exerciseinferencet.mat';
clearvars -except lwage IQ exper educ age agesquared;
```

2.2. Create the systematic component of the regression

```
y = lwage;
N_obs = size(y,1);
x_0 = ones(N_obs,1);
X = [x_0 IQ exper educ];
```

2.3. Obtain the OLS coefficient estimate of interest and the S.E. estimate of it

The supplied MATLAB function file calculates OLS statistics. We are interested in the OLS coefficient estimate of IQ, and the standard error estimate of it. Study in this function file how OLS coefficient estimates and their standard error estimates are calculated. Obtain the regarding estimates in your script file using the code presented at the end of the section.

Observe that the coefficient estimate is 0.0058. The standard error estimate of this coefficient estimate is 0.00097966.

```
LSS = exercisefunction(y,X);
B_hat_k = LSS.B_hat(2,1);
```

```
B_hat_k_SEE = LSS.B_hat_SEE(2,1);
```

3. Determine the test statistic for the hypothesis test on the population coefficient

3.1. Hypothesised value of the true coefficient

Suppose that someone claims that each additional IQ point raises one's wage by 0.75 percent on average. This means that the hypothesised population coefficient is $\beta_k^0 = 0.0075$ where $k = \text{IQ}$. To test the claim against the alternative, we would then formulate the null hypothesis as $H_0 : \beta_k = 0.0075$, and the alternative hypothesis as $H_1 : \beta_k \neq 0.0075$.

```
B_k_0 = 0.0075; % Typically this is 0 but not in this example.
```

3.2. Determine the test statistic, state its distribution derived under the null hypothesis, and calculate it

We want to test whether the coefficient estimate of IQ is statistically different from the hypothesised value. We can use the t statistic to carry out this hypothesis test. Under the null hypothesis, and the OLS assumptions, the test statistic follows a t distribution with a certain degrees of freedom to be calculated below. We can calculate the t value using the coefficient estimate on the independent variable of interest, the hypothesised value of the population coefficient, and the standard error estimate of this coefficient estimate obtained above. The code at the end of the section calculates the t value as -1.70.

```
t = (B_hat_k-B_k_0)/B_hat_k_SEE; % t statistic has a t distribution.
t = round(t,2); % This is an adjustment for visualisation later in the exercise.
```

3.3. Determine the degrees of freedom of the t distribution

The t distribution depends on one parameter. This parameter is the degrees of freedom. In the regression context, the degrees of freedom is equal to the number observations less the number of coefficients, including the intercept, to estimate.

In the code stated at the end of the section, `size(X,2)` determines the number of coefficients to estimate. In particular, the built-in `size` function takes two input arguments. The first input argument is the matrix of interest, and the second input argument is the indicator of the matrix dimension of interest. The value 2, in particular, indicates that it is the column dimension of the matrix that is of interest. A value of 1 would have indicated the row dimension of the matrix. We are interested in the size of the column dimension of the matrix because this gives the number of coefficients we are estimating.

```
t_df = N_obs-size(X,2);
```

3.4. Calculate the p value corresponding to the t value

Another parameter we need to calculate to conduct the t test is the probability value of the test. The probability value is referred to as the p value.

Before calculating the p value, first note that we have formulated our null hypothesis as whether $\beta_k = \beta_k^0$ where β_k^0 is hypothesised to be equal to 0.0075. Since the difference $\beta_k - \beta_k^0$ can be negative or positive, the t test of interest is a two-tailed test. This means that we need to calculate the p value such that the probability area covers both tails of the t distribution. In Section 5 of this exercise, what this probability area is about will become much more clear because we will work with the t distribution itself right in front of us. For the moment, keep calm and calculate your p value.

We can use standard statistical software to calculate the p value. Here, in particular, we can use the built-in MATLAB function `tcdf` to obtain the tail probability given the t value calculated above, and the degrees of freedom of the t distribution. The p value is equal to 0.0805.

Note that the particular p value we have calculated using MATLAB cannot be obtained using the tabulated t distribution at the back of your textbook. This tabulation does not present all probability values corresponding to all possible t values for space reasons. It gives the impression that it does. It does not. This fact makes the understanding of the t test, or any hypothesis test, difficult when someone is learning hypothesis testing the first time. We will return back to this later in the exercise.

```
p = (1-tcdf(abs(t),t_df))*2; % The test is two-tailed.
```

4. Carry out the two-tailed t test using the tabulated t distribution

4.1. Calculate the critical t value corresponding to a critical p value

Assume a significance level of 0.05. Let us refer the 'significance level' as the 'critical p value'. Given this critical p value and the degrees of freedom of the test, we need to calculate the 'critical t value'. For this, you would typically use the tabulated t distribution presented at the end of a statistic book. From where these tabulated values come from is a mystery we will solve in Section 5. Here we use the built-in MATLAB function `tinv` to obtain the critical t value. You may want to type `doc tinv` in the command prompt to learn about the input arguments of the `tinv` function, and the output arguments it returns.

Note that the critical p value needs to be divided by 2 when obtaining the critical t value for the lower or the upper tail of the t distribution since the test of interest is a two-tailed test.

Compare the t value with the critical t value. What do you conclude?

```
t_c = tinv(0.025,t_df); % The test is two-tailed.  Here critical p value refers
to the significance level.  Choose this as 0.05.  Fail to reject the null since
t > t_c.
t_c = round(t_c,2); % This is an adjustment to better visualise a plot later in
the exercise.
```

4.2. Calculate the critical p value corresponding to the critical t value

Consider the `tcdf` function to obtain the two-tail probability given the critical t value and the degrees of freedom of the t distribution. What do you conclude?

```
p_c = (1-tcdf(abs(t_c),t_df))*2; % The test is two-tailed.  Fail to reject the
null since p > p_c.
```

5. Carry out the two-tailed t test using the simulated t distribution

In the preceding sections we have calculated the t value, the critical t value, the p value, and the critical p value to conduct our hypothesis test. We have obtained these parameters using built-in MATLAB functions. If we did not have access to MATLAB, we would be using the tabulated distribution of the test statistic at the back of a statistics textbook. Note that all these parameters, and therefore the hypothesis test itself, regards the distribution of the test statistic. However, we have never seen the distribution itself while conducting our hypothesis test! Therefore, it is not very clear how we have been conducting the test. O'Hara (International Review of Economics Education, 2018) proposes that instructors move away from using the tabulated distribution of the test statistic at the back of the textbooks when teaching hypothesis testing. Instead, he proposes that instructors teach students to test hypotheses by using the simulated distribution of the test statistic which can be created using random number generators in statistical software. This provides students with a visual and intuitive understanding of the sampling distribution, and the logic behind hypothesis testing. We will now pursue the approach of O'Hara to learn about conducting the t test.

5.1. Set the number of theoretical observations for the data to be used to simulate the t distribution

Set the number of theoretical observations for the data to be used to simulate the t distribution.

```
N_obs_data_sim_t_dis = 100000; % What do you expect will happen if you increase
this number?  Check this after you complete the exercise.
```

5.2. Generate a dataset of theoretical observations of a random var. with the distribution and deg. of freedom the test statistic would follow under the null hypothesis

The test statistic of interest is the t statistic. A test statistic is a random variable. The t statistic is a random variable, and it follows a t distribution with a given degrees of freedom. Our aim is to draw theoretical observations for a random variable that follows the t distribution. We call these observations 'theoretical observations' because each observation is a random variable with a probability distribution. They are not 'realised observations' of some given data. They are theoretical observations of random data. Hardly any textbook discusses

this distinction. One exception is Magnus (2017, Introduction to the Theory of Econometrics).

Drawing theoretical observations from an underlying distribution is easy to do with the built-in MATLAB function `trnd`. The `trnd` function accepts two input arguments. The first input argument is the degrees of freedom. This is an obvious argument we would expect the `trnd` function to accept because the degrees of freedom is a parameter of the t distribution.

The second input argument of the `trnd` function is for the dimension of the matrix that will contain the random draws from the t distribution. We want a column of random numbers with `N_obs_data_sim_t_dis` theoretical observations. Therefore, we specify the row dimension of the matrix as `N_obs_data_sim_t_dis`, and the column dimension of the matrix as `1`.

```
data_sim_t_dis = trnd(t_df,[N_obs_data_sim_t_dis,1]); % The random variable of
interest has a t distribution.  Each time trnd is called, different random
draws are taken from the t distribution.
```

5.3. Set the number of bins for the histogram to be drawn

`nbins` is an input argument of the built-in `histogram` function. It specifies the number of bins of a histogram. The bins divide the entire range of values into a series of intervals. If a number of bins is not specified as an input argument, then `histogram` automatically calculates how many bins to use based on the values underlying the histogram.

Below we will produce histograms and overlay them to obtain one histogram. We need to fix the number of bins across these histograms so that we can interpret the final histogram with no difficulties.

In the code, `nbins = -5:0.01:5`, we define the number of bins as a range. The range starts from `-5` and runs until `5` with increments of `0.01`. The end points of the range regard the least frequent values any t distribution can take at its tails regardless of the degrees of freedom parameter of the t distribution. We divide the entire range of values into intervals of size `0.01` because this results in a histogram that is visually easy to interpret.

```
nbins = -5:0.01:5;
```

5.4. Plot the histogram of the t distribution

The `histogram` function presented at the end of the section produces a histogram based on the dataset `data_sim_t_dis` generated above. The data contains the theoretical observations of the random variable following the t distribution. This histogram is an approximation of the continuous t distribution, and provides us with enough visual information to study the t test.

```
histogram(data_sim_t_dis,nbins,'FaceColor','white','EdgeAlpha',0.15);
hold off
title('Fig.  1:  Simulated t distribution')
legend('t distribution')
ylabel('Frequency')
xlabel('t')
```

5.5. Mark the t value

On the histogram you just plotted, mark where the t value you calculated from the wage data would fall in this distribution. Does this seem like a likely value to observe if in fact this is the true distribution?

Having this distribution right in front of us allows us to make the connection between the t value we have computed from the data, and the distribution to which we are to compare it. We can see that, although the value of the test statistic from the data is toward the lower tail, it is not all that unlikely as an outcome from this distribution. This is much more intuitive than comparing the value to a bunch of numbers in the table at the back of a textbook.

```
histogram(data_sim_t_dis,nbins,'FaceColor','white','EdgeAlpha',0.15);
hold on
line([t t],ylim,'Color','blue') % Mark t value.  Does this seem like a likely
value to observe if in fact this is the true distribution?
hold off
title('Fig.  2:  Simulated t dis.  with t value marked')
legend('t distribution','t value')
ylabel('Frequency')
xlabel('t')
```

5.6. Shade the areas where the random values are more extreme than the absolute value of the t value

The code at the end of the section shades the areas of the histogram where the random values are more extreme than the absolute value of the t value. We consider the absolute value, that is both tails of the distribution, because the test of interest is two-tailed. The meaning of the shaded areas is the topic of the next section.

```
val_below_abs_t = data_sim_t_dis <= -abs(t);
val_above_abs_t = data_sim_t_dis >= abs(t);
val_between = data_sim_t_dis > -abs(t) & data_sim_t_dis < abs(t);
histogram(data_sim_t_dis(val_below_abs_t),nbins,'FaceColor','blue','EdgeAlpha',0.15);
% The corresponding area is p_sim/2.  See below.
hold on
histogram(data_sim_t_dis(val_above_abs_t),nbins,'FaceColor','blue','EdgeAlpha',0.15);
% The corresponding area is p_sim/2.  See below.
hold on
histogram(data_sim_t_dis(val_between),nbins,'FaceColor','white','EdgeAlpha',0.15);
hold on
line([t t],ylim,'Color','blue')
hold off
title('Fig.  3:  Simulated t dis.  with t value marked and prob.  areas shaded')
legend('Values below -abs(t value)','Values above abs(t value)','Values between','t
value')
```

```
ylabel('Frequency')
xlabel('t')
```

5.7. Calculate the shaded area associated with the t value which gives the simulated p value

Calculate a variable called `t_extreme_value_dummy_t` that takes a value of 1 when the absolute value of the randomly generated number stored in `data_sim_t_dis` is greater than the absolute value of the test statistic calculated from the wage data, and a value of 0 otherwise.

Next, calculate the proportion of the time `t_extreme_value_dummy_t` takes a value of 1 to the whole. This can be calculated by simply calculating the mean of the extreme variable, since this will be adding up all the 1 values and dividing by the total number of random values `100000` defined above. This mean is approximately 0.0800; it depends on the random numbers generated above. This is the probability that a randomly generated value from the simulated t distribution is above the t value calculated from the wage data. It is the simulated p value for the hypothesis test!

Compare `p_sim` with `p` calculated earlier in this exercise. The two quantities are close to each other. This is not surprising. `p_sim` is the probability value of the test based on the simulated t distribution, and `p` is the probability value of the test based on the continuous t distribution. If you increase `N_obs_sim_t_dis_data`, `p_sim` will start to converge to `p`.

```
t_extreme_value_dummy_t = abs(data_sim_t_dis) > abs(t);
p_sim = mean(t_extreme_value_dummy_t); % The fraction of the extreme values
gives the probability area associated with the t value under the simulated
t distribution.  This gives the simulated p value!  Compare p_sim with p!  If
you increase N_obs_sim_t_dis_data,p_sim gets closer to p.  Why?
```

5.8. Mark the t value, mark the critical t value, shade the areas where the random values are more extreme than the absolute value of the t value, and shade the areas where the random values are more extreme than the absolute value of the critical t value

The code at the end of the section marks where the critical t value calculated from the t distribution earlier in this exercise would fall in this distribution. Furthermore, it shades, in red colour, the areas of the histogram where the random values are more extreme than the absolute value of the critical t value. The meaning of the shaded areas associated with the critical t value is the topic of the next section.

Having the simulated t distribution in front of us, and having the value of the test statistic, the critical value of the test statistic, and the associated probability areas marked on the simulated distribution, we can now conduct our hypothesis test. Looking at the lower tail of the simulated t distribution, the t value lies above the critical t value. That is, `t > t_c`. Accordingly, the probability area associated with the t value is larger than the probability area associated with the critical t value. That is, `p_sim > p_c`. Therefore, we fail to reject the null hypothesis. Remember that `p_c` is just your choice of a significance level.

```
val_below_abs_t_c = data_sim_t_dis <= -abs(t_c);
val_above_abs_t_c = data_sim_t_dis >= abs(t_c);
```

```
histogram(data_sim_t_dis(val_below_abs_t),nbins,'FaceColor','blue','EdgeAlpha',0.15);
hold on
histogram(data_sim_t_dis(val_below_abs_t_c),nbins,'FaceColor','red','EdgeAlpha',0.15);
hold on
histogram(data_sim_t_dis(val_above_abs_t),nbins,'FaceColor','blue','EdgeAlpha',0.15);
hold on
histogram(data_sim_t_dis(val_above_abs_t_c),nbins,'FaceColor','red','EdgeAlpha',0.15);
% Fail to reject the null since p_sim > p_c.
hold on
histogram(data_sim_t_dis(val_between),nbins,'FaceColor','white','EdgeAlpha',0.15);
hold on
line([t t],ylim,'Color','blue')
hold on
line([t_c t_c],ylim,'Color','red') % Mark the critical t value.  Fail to reject
the null since t > t_c.
hold off
title('Fig.  4:  Simulated t dis.  with t and critical t values marked,
prob.  areas shaded')
legend('Values below -abs(t value)','Values below -abs(critical t value)',
'Values above abs(t value)','Values above abs(critical t value)',
'Values between','t value','critical t value')
ylabel('Frequency')
xlabel('t')
```

5.9. Calculate the shaded area associated with the critical t value which gives the simulated critical p value

Calculate a variable called `t_extreme_value_dummy_t_c` that takes a value of 1 when the absolute value of the randomly generated number stored in `data_sim_t_dis` is greater than the absolute value of the critical value of the test statistic calculated from the t distribution, and a value of 0 otherwise.

Next, calculate the proportion of the time `t_extreme_value_dummy_t_c` takes a value of 1 to the whole. This can be calculated by simply calculating the mean of the extreme variable, since this will be adding up all the 1 values and dividing by the total number of random values defined above. This mean is approximately 0.0500. This is the probability that a randomly generated value from the simulated t distribution is above the critical t value calculated from the t distribution. It is the simulated critical p value for the hypothesis test.

Compare `p_c_sim` with `p_c` calculated earlier in this exercise. The two quantities are close to each other. This is not surprising. `p_c_sim` is the critical probability value of the test based on the simulated t distribution, and `p_c` is the critical probability value of the test based on the continuous t distribution. If you increase `N_obs_sim_t_dis_data`, `p_c_sim` will start to converge to `p_c`.

```
t_extreme_value_dummy_t_c = abs(data_sim_t_dis) > abs(t_c);
p_c_sim = mean(t_extreme_value_dummy_t_c); % The fraction of the extreme values
gives the probability area associated with the critical t value under the
```

```
simulated t distribution.  This gives the simulated critical p value!  Compare
p_c_sim with p_c!  If you increase N_obs_sim_t_dis_data, p_c_sim gets closer to
p_c.  Why?
```

§ Humor

> "Apply the *laugh* test. If the findings were explained to a layperson, could that person avoid laughing?"