Problem 1

Remember from last week we discussed that skewness and kurtosis functions in statistical packages are often biased. Is your function biased? Prove or disprove your hypothesis.

Solution:

The function used are scipy.stats.skew and scipy.stats.kurtosis(). By definition, these two functions are default biased. As both of the functions have a parameter bias.

Text

Description automatically generated

We solve this problem by doing the hypothesis test.

Steps

1. Sample 10, 100, 1,000, 10,000 standardized random normal values.

2. Calculate the skewness and kurtosis by python function skew() and kurtosis().

3. Sample the skewness and kurtosis by repeating steps 1 and 2 100 times.

4. Calculate the p-value (μ0 = 0).

5. If the value is lower than your threshold (typically 5%), then you reject the hypothesis that the kurtosis function is unbiased.

We have the following result:

Text

Description automatically generated

For **kurtosis()**, it is **biased** as we fail to rejected the null hypothesis at significant level of 0.05 under all sample size tested.

As sample size going larger and larger, the different between biased and unbiased will become smaller and smaller. So we focus more on the smaller sample size case.

For

Problem 2

2.1 Fit the data in problem2.csv using OLS and calculate the error vector. Look at its distribution. How well does it fit the assumption of normally distributed errors?

Solution:

Using OLS regression by using the python sm.OLS() function. The OLS Regression results is shown by ols\_model.summary() as

Graphical user interface, text

Description automatically generated

The error vector is calculated by ols\_model.resid. Part of the error vector is shown as following:

Calendar

Description automatically generated with medium confidence

The 4 moments of the distribution is calculated. The histogram of distribution is also plotted.

Chart, histogram

Description automatically generated

The Skewness of error is -0.26726658552879606. The Kurtosis of error is 3.1931010009568777. The Std of error is 1.2044314159651845. These three moments is far from that of the normal distribution. The histogram of it is not like normal distribution as well. It does not fit the assumption of normally distributed errors.

2.2 & 2.3

Fit the data using MLE given the assumption of normality. Then fit the MLE using the assumption of a T distribution of the errors. Which is the best fit?

What are the fitted parameters of each and how do they compare? What does this tell us about the breaking of the normality assumption in regards to expected values in this case?

The Goodness of fitness statistic R Square and Information criteria AIC and BIC is used in model selection.

|  |  |  |  |
| --- | --- | --- | --- |
|  | OLS | MLE\_Norm | MLE\_T |
| Const | 0.1198 | 0.1198 | 0.1426 |
| X | 0.6052 | 0.6052 | 0.5576 |
| R\_Squared | 0.195 | 0.1946 | 0.1934 |
| AIC | 324.0 | 324.0 | 315.0 |
| BIC | 329.2 | 329.2 | 320.2 |

The OLS and the MLE with normal distribution led to the same result. Then MLE\_Norm and MLE\_T are compared. The first model selection statistic R\_Squared is quite similar. But the MLE\_T has smaller AIC and BIC. The MLE\_t is the best fit.

Both OLS and MLE\_Norm based on the assumption of normal distribution. From this result we can see that the normality assumption might be broken. And use the MLE with t distribution might have a better result.

Problem 3

Graphical user interface

Description automatically generated

Graphical user interface

Description automatically generated

The simulate process are drawn as labeled in the graph.

|  |  |  |
| --- | --- | --- |
|  | ACF | PACF |
| AR | Decrease exponentially | Rapid decrease after order |
| MA | Rapid decrease after order | Decrease exponentially |

* Decrease refers to absolute value

By the pattern of decrease rapidly we can identify the type and order of each process.