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.

Answer:

The function `kurtosis()` used in Python statistical package `scipy.stats` is default biased. But it could be changed by adding parameter “`bias = False`”. But the calculation of skewness, function `skew()`, although biased, is so small that it cannot be proven to be biased at a significance level of 0.05.Here's how I proved this thing.

To prove this hypothesis, I first generated several sets of data with known population distribution, which is the standard normal distribution. The kurtosis and skewness of each set of data were then calculated and collected separately, using functions in Python. Hypothesis testing is then performed.

The null hypothesis I made is that the results obtained by the function calculation are unbiased, that is to say, the calculated values should all be equal to zero, for normally distributed data. And the alternative hypothesis is that neither of these two data is equal to zero. And set the significance level as 0.05. By calculating T statistic, the following results are obtained:

Sample Number	Skewness		Kurtosis	
	p-value	Reject	p-value	Reject
100	0.860	Fail	0.000	Reject
1,000	0.596	Fail	0.951	Fail
10,000	0.134	Fail	0.849	Fail

It can be clearly seen that the calculation of kurtosis is biased, and this deviation cannot be ignored when the sample size is small, but as the sample size increases, this result will gradually approach the unbiased calculation result.

For the calculation of skewness, when the parameter "bias" is not changed, Python uses a simplified calculation formula:

$$Skewness = \frac{3\ (Mean - Median)}{Standard\ Deviation}$$

Rather than the calculation formula of skewness that we usually think of. It may be more rigorous to draw conclusions like this: in principle, although this function provides an estimate of the biased skewness calculation, the result cannot be proved to be biased for randomly generated data that conforms to the standard normal distribution, at a significance level of 0.05.

Problem 2

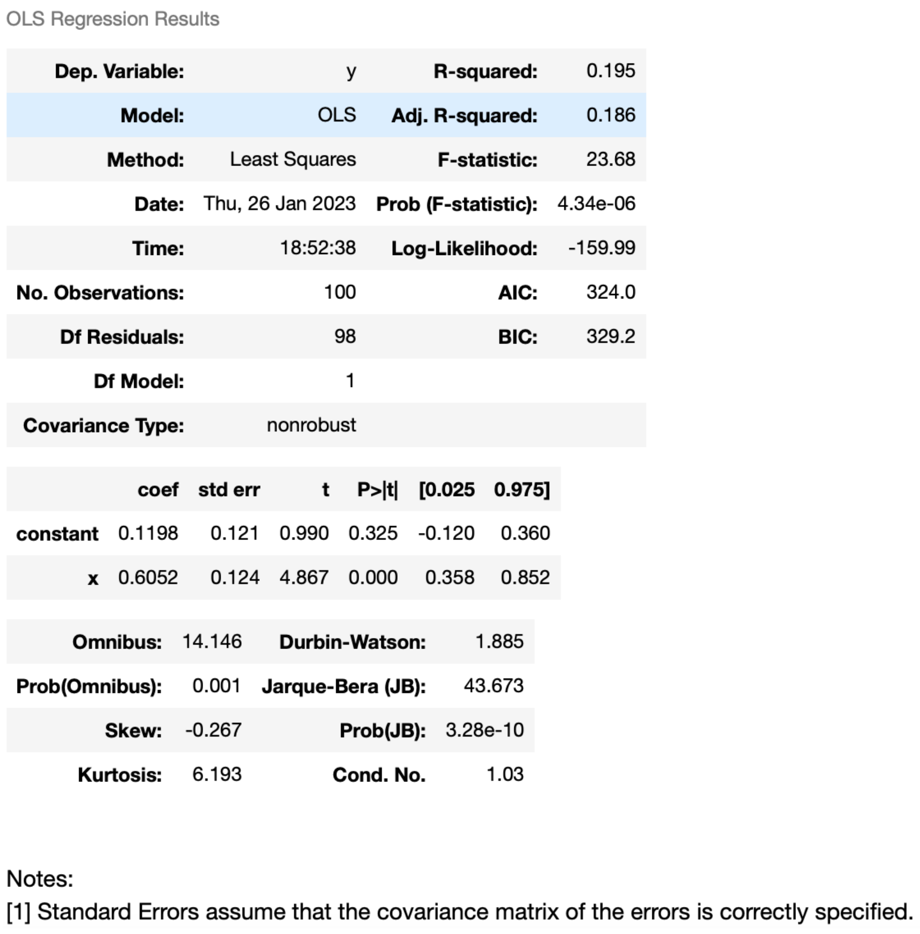
Fit the data in problem2.csv using OLS and calculate the error vector. Look at its distribution. How well does it fit the assumption of normally distributed errors?

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?

Answer:

For problem2.csv using OLS fitting, the results obtained are as follows:



OLS Regression Results

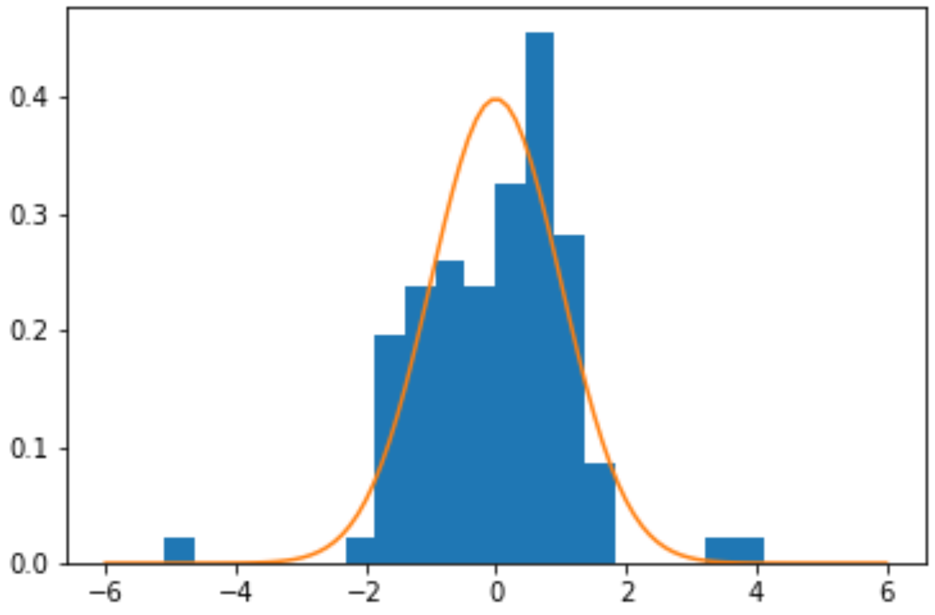
Dep. Variable:	y	**R-squared:**	0.195
Model:	OLS	**Adj. R-squared:**	0.186
Method:	Least Squares	**F-statistic:**	23.68
Date:	Thu, 26 Jan 2023	**Prob (F-statistic):**	4.34e-06
Time:	18:52:38	**Log-Likelihood:**	-159.99
No. Observations:	100	**AIC:**	324.0
Df Residuals:	98	**BIC:**	329.2
Df Model:	1		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
constant	0.1198	0.121	0.990	0.325	-0.120	0.360
x	0.6052	0.124	4.867	0.000	0.358	0.852

Omnibus:	14.146	**Durbin-Watson:**	1.885
Prob(Omnibus):	0.001	**Jarque-Bera (JB):**	43.673
Skew:	-0.267	**Prob(JB):**	3.28e-10
Kurtosis:	6.193	**Cond. No.**	1.03

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

The statistical value of the residual item is calculated, and the skewness is -0.27, and the kurtosis is 3.19, which has a large deviation from the normal distribution. Comparing its distribution graph with the probability distribution curve of the normal distribution (see the figure below), it can also be clearly seen that there is a large deviation. That is, it deviates considerably from the assumption of a normal distribution.



Based on the assumptions of normal distribution and t-distribution, the results calculated by MLE are as follows:

	Normal Distribution	**t – Distribution**
Intercept	0.1198	0.1426
Beta	0.6052	0.5576
R-square	0.1946	0.1934
AIC	323.98	314.95
BIC	329.19	320.16

From the above results, it can be seen that based on the assumption of normal distribution, the calculation results of MLE are exactly the same as those of OLS. This conclusion fits with our class notes.

In order to compare which of these two models fit better, I calculated their R-square, AIC and BIC respectively. Since the results given by R-square are relatively close, no comparison can be made. So we will compare according to Information Criteria.

Since the AIC and BIC values of the model based on the t-distribution are both lower, this model is the better model.

This result tells us that for a set of data, although OLS usually gives us good results, because it is based on the assumption that the residuals conform to a normal distribution, this assumption is usually likely to be broken. In this case, we can make new assumptions about the residuals and use MLE to fit them, and the results obtained may be better.

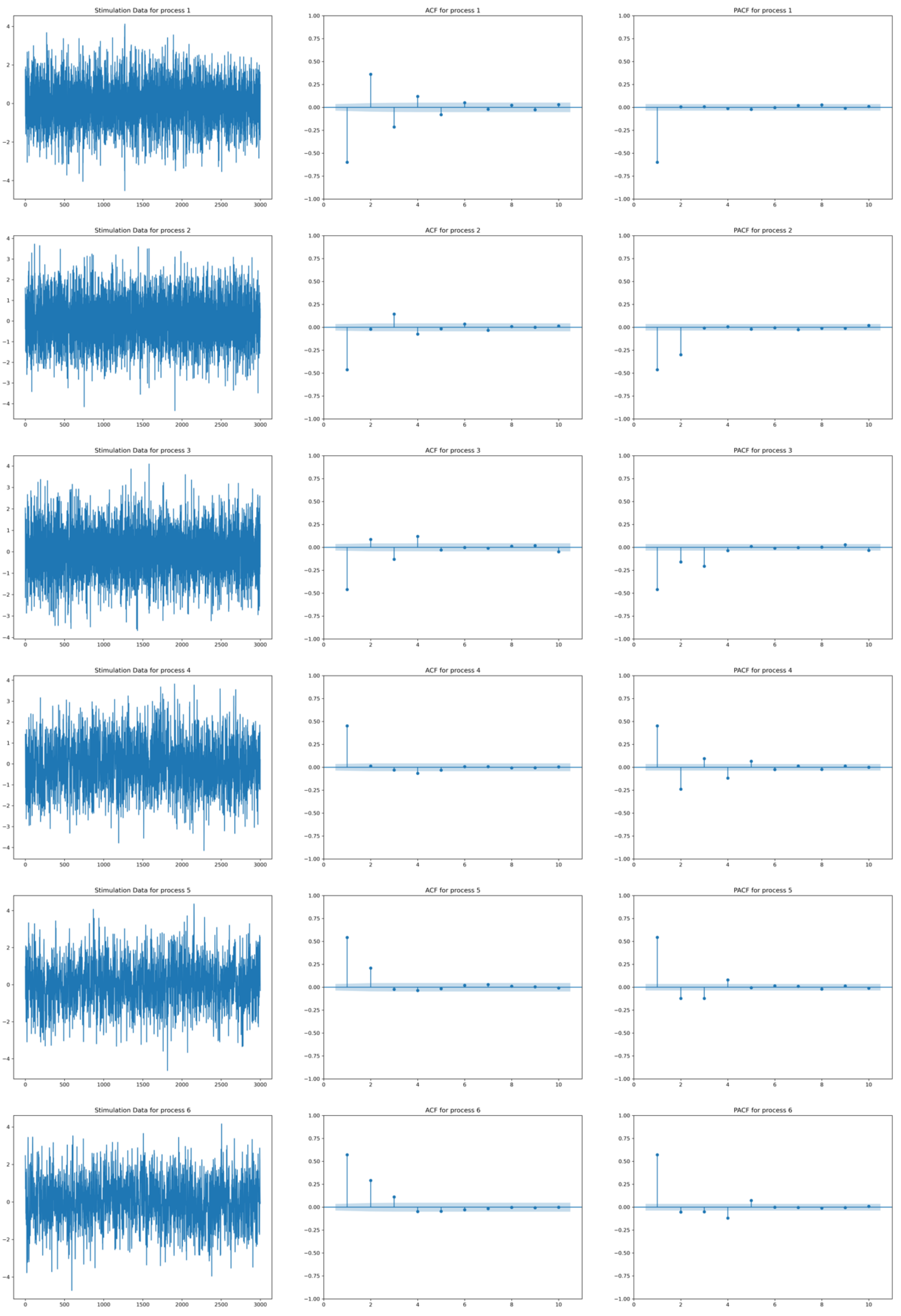
Problem 3

Simulate AR(1) through AR(3) and MA(1) through MA(3) processes. Compare their ACF and PACF graphs. How do the graphs help us to identify the type and order of each process?

Answer:

The results obtained using the python simulation are on the next page. Among them, the first three lines correspond to the simulation results of AR(1) through AR(3), and the last three lines correspond to the process of MA(1) through MA(3).

The graphs help us to identify the type and order of the process by the shape of the plots. For an AR process, the ACF will decay exponentially while the PACF will cut off after the lag corresponding to the order of the process. For an MA process, the PACF will decay exponentially while the ACF will cut off after the lag corresponding to the order of the process. So by comparing the ACF and PACF graph, we can identify the type and order of each process.


Stimulation Data for process 1
ACF for process 1
PACF for process 1
Stimulation Data for process 2
ACF for process 2
PACF for process 2
Stimulation Data for process 3
ACF for process 3
PACF for process 3
Stimulation Data for process 4
ACF for process 4
PACF for process 4
Stimulation Data for process 5
ACF for process 5
PACF for process 5
Stimulation Data for process 6
ACF for process 6
PACF for process 6