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.

Notes

The sample skewness is computed as the Fisher-Pearson coefficient of skewness, i.e.

$$g_1 = \frac{m_3}{m_2^{3/2}}$$

where

$$m_i = \frac{1}{N}\sum_{n=1}^{N}(x[n] - \bar{x})^i$$

is the biased sample $i\texttt{th}$ central moment, and $\bar{x}$ is the sample mean. If `bias` is False, the calculations are corrected for bias and the value computed is the adjusted Fisher-Pearson standardized moment coefficient, i.e.

$$G_1 = \frac{k_3}{k_2^{3/2}} = \frac{\sqrt{N(N-1)}}{N-2}\frac{m_3}{m_2^{3/2}}.$$

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 ($\mu 0 = 0$).
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:

```
10 samples:
skewness p-value is 0.010022125067440038.
Reject the null hypothesis.

kurtosis p-value is 9.892089260419227e-12.
Reject the null hypothesis.

100 samples:
skewness p-value is 0.835437664512038.
Fail to reject the null hypothesis.

kurtosis p-value is 0.6467761933137416.
Fail to reject the null hypothesis.

1000 samples:
skewness p-value is 0.0375664513602751.
Reject the null hypothesis.

kurtosis p-value is 0.9977733631042587.
Fail to reject the null hypothesis.

10000 samples:
skewness p-value is 0.472536613094932.
Fail to reject the null hypothesis.

kurtosis p-value is 0.05100055903843119.
Fail to reject the null hypothesis.
```

These two functions are **biased** when sample size is small as we rejected the null hypothesis at significant level of 0.05 with 10 samples tested.

As sample size going larger and larger, the functions tend to be unbiased.

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

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:	Fri, 27 Jan 2023	Prob (F-statistic):	4.34e-06
Time:	21:27:53	Log-Likelihood:	-159.99
No. Observations:	100	AIC:	324.0
Df Residuals:	98	BIC:	329.2
Df Model:	1		
Covariance Type:	nonrobust		

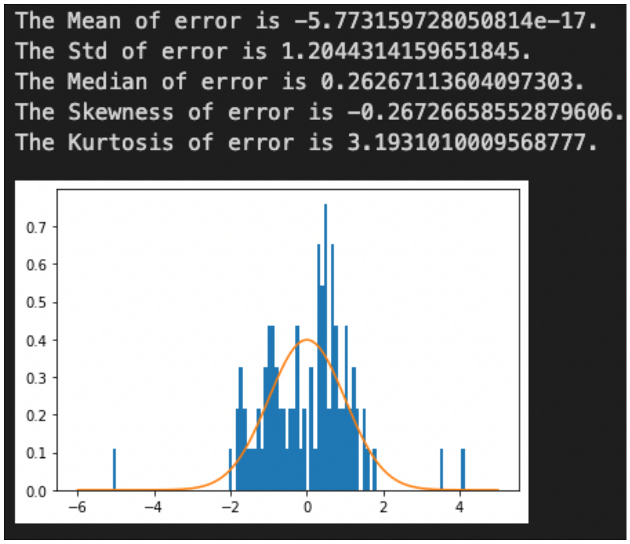
	coef	std err	t	P>|t|	[0.025	0.975]
ones	0.1198	0.121	0.990	0.325	-0.120	0.360
x	0.6052	0.124	4.867	0.000	0.358	0.852

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

The error vector is calculated by ols_model.resid. Part of the error vector is shown as following:

```
0    -0.838485
1     0.835296
2     1.027428
3     1.319711
4    -0.152317
        ...
95   -1.590264
96   -1.694848
97    0.434878
98    0.402261
99   -0.922319
```

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



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?

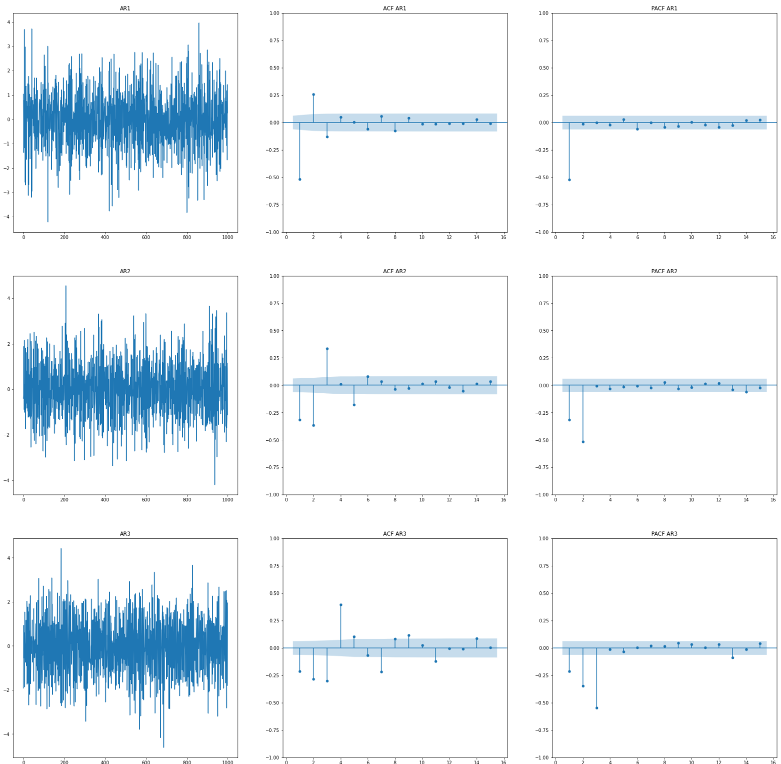
The Goodness of fitness statistic R Square and Information criteria AIC and BIC is used in model selection.

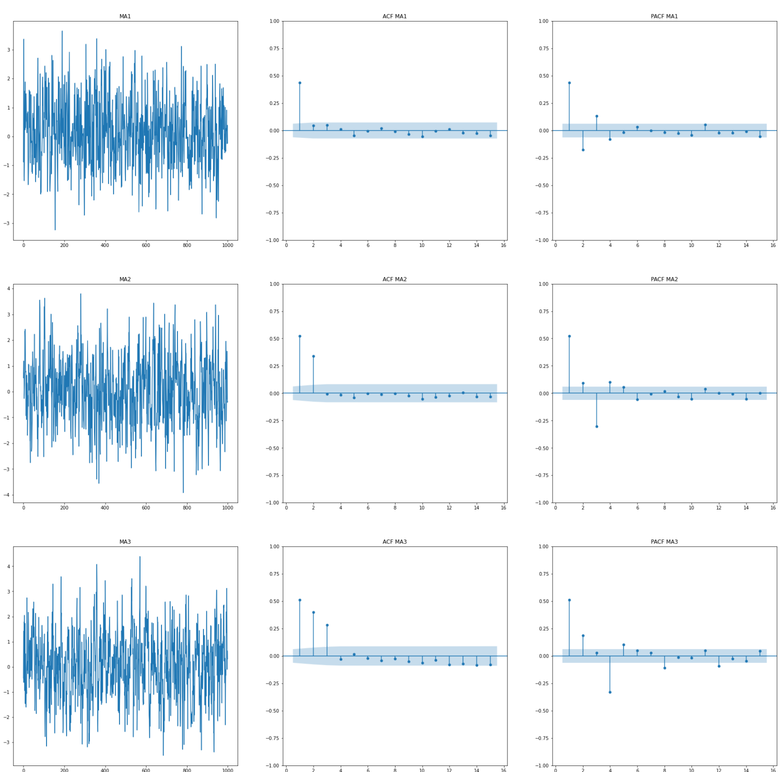
	OLS	MLE_Norm	MLE_T
Const	0.1198	0.1198	0.1426
X	0.6052	0.6052	0.5576
R_Squared	0.195	0.1946	0.1934
AIC	324.0	324.0	315.0
BIC	329.2	329.2	320.2

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


AR1
ACF AR1
PACF AR1
AR2
ACF AR2
PACF AR2
AR3
ACF AR3
PACF AR3


MA1
ACF MA1
PACF MA1
MA2
ACF MA2
PACF MA2
MA3
ACF MA3
PACF MA3

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.