

Course: FINTECH 545
Name: Miranda Lu
Title: Week2 Assignment

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:

Following the instruction of the course notes, I sampled 10 standard normally distributed random variables to calculate the skewness and the kurtosis, and repeated this step for 1000 times with the *scipy* package in Python. The results are as follows:

Skewness:

```
Skewness Test:  
T value of skewness is 1.2184699218115365, the p value of skewness is 0.22333311454388746  
So, insignificantly biased
```

Kurtosis:

```
Kurtosis Test:  
T value of kurtosis is -23.051155894849412, the p value of kurtosis is 1.288524162962821e-94  
So, significantly biased
```

The results shows that we cannot reject the hypothesis that the skewness equals to zero, while we can reject the hypothesis that the kurtosis equals to 3.

The results imply that the **SKEWNESS** calculated by **scipy** package is **UNBIASED**, while the **KURTOSIS** calculated by **scipy** package is **BIASED**.

According to the formula *scipy* used, it tends to overestimate the skewness of a dataset, especially for smaller sample sizes. To correct for this bias, we need to use the unbiased estimator of skewness. The corrected formula is :

$$\text{skewness_corrected} = \text{skew}(\text{data}) * \sqrt{((n-1)/(n-2))}$$

And the kurtosis follows a similar pattern.

Thus, the result of kurtosis meets the expectation, while the skewness does not. It might be results the number of sample is not small enough or we need repeat the step more times. Bad luck is not contributing to this results as I've tried multiple times.

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?

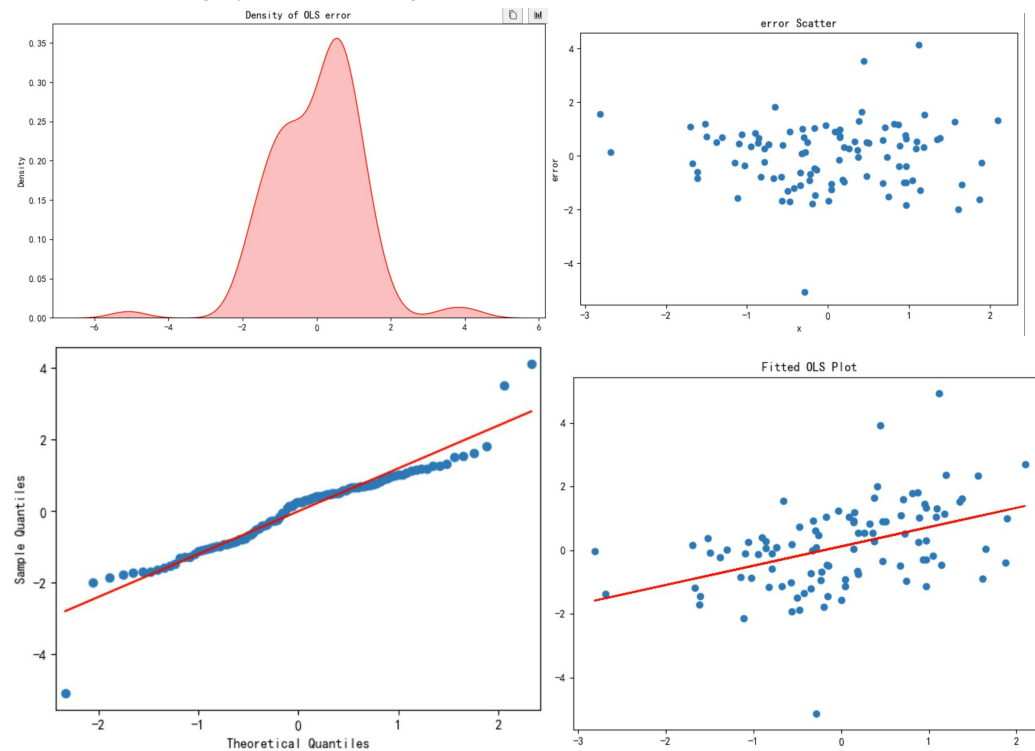
Fit the data using MLE given the assumption of normality. Then fit the MLE using the assumption of a T distribution of the errors. Which is the best fit?

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:

The error vector produced by OLS is **NOT** normally distributed.

The distribution graphs are showing as follows:



We can tell from these graphs that the error vector is not normally distributed..

The hypothesis testing with H_0 being normally distributed error terms shows the following results,

```
The result of using normaltest:
stats = 14.146365351261274, and the pvalue = 0.0008475313945909519
The error vector is not normally distributed
=====

The result of using kstest:
stats = 14.146365351261274, and the pvalue = 0.0008475313945909519
The error vector is not normally distributed
=====

The result of using Anderson-Darling:
stats = 14.146365351261274, and the pvalue = 0.0008475313945909519
The error vector is not normally distributed
=====

The result of using Shapiro-Wilk test:
stats = 14.146365351261274, and the pvalue = 0.0008475313945909519
The error vector is not normally distributed
```

Several statistical methods are employed, including the Anderson-Darling test and the Shapiro-Wilk test, all of which shows that the error terms are statistically significantly different

from normal distribution.

With normality assumption, calculating the linear regression coefficients with MLE reached the following results,

```
Given the assumption of normal distribution, MLE parameters are: [0.60520484 0.11983623 1.1983941 ]
Given the assumption of t-distribution, MLE parameters are: [0.55757182 0.14261399 6.27658083 0.97126631]
```

The coefficients with t distribution are also shown above.

I calculate the positive log-likelihood function with the PDF of each distribution, and use `scipy.optimize.minimize()` to find the parameters minimizing the absolute value of log-likelihood function. The coefficients for normal distribution are slope, intercept, and sigma, respectively. Those for t distributions are slope, intercept, degree of freedom, and scale, respectively.

To determine the most suitable model, I use AIC, AICc, and BIC, where each of these indicators show that model with **t distribution assumption outperforms the one with normal distribution assumption**. The detailed results are

```
if Normal distribution, AIC = 325.98419337832485 , AICc = 326.23419337832485 , BIC = 333.7997039362891
if t distribution, AIC = 318.94594082499174 , AICc = 328.4052460099038 , BIC = 338.4048741222772
According to AICc, t distribution performs better
According to BIC, t distribution performs better
```

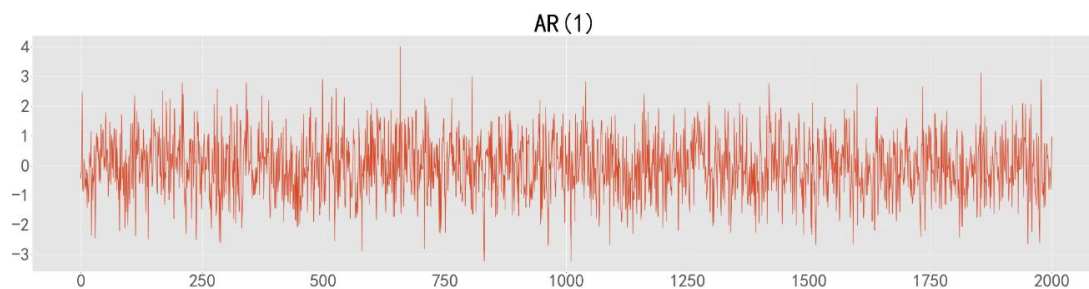
As shown above, model with t distribution has a lower intercept and steeper slope. If the normality is broken, then OLS estimators are no longer the Best Linear Unbiased Estimator, so the expected value of OLS estimators no longer equal to the true population parameters. However, both OLS and MLE can still produce consistent estimators.

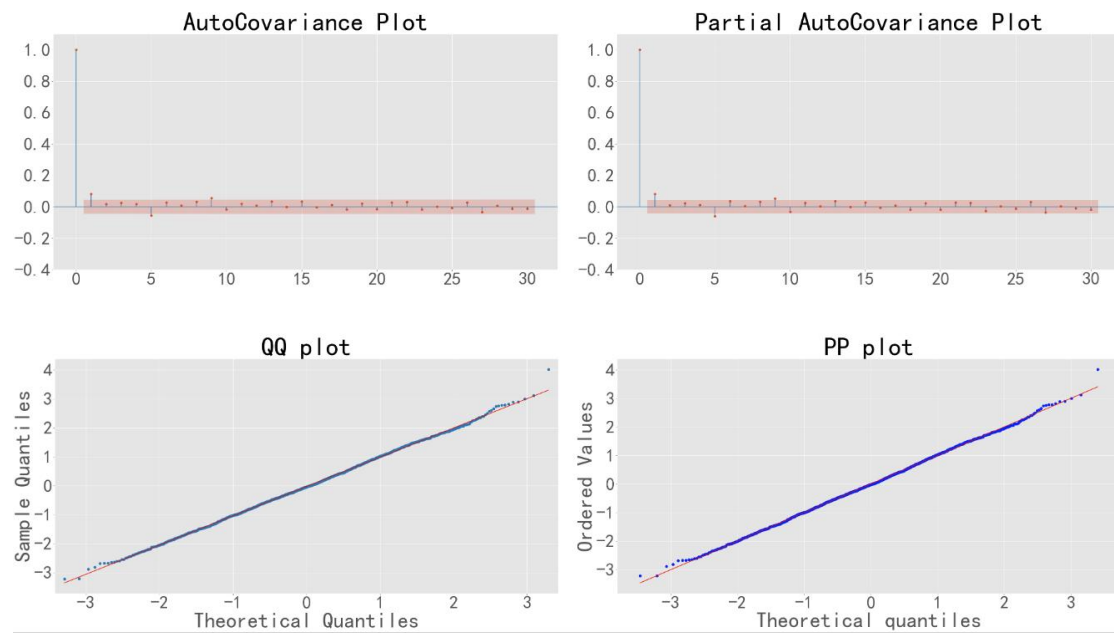
Problem 3

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

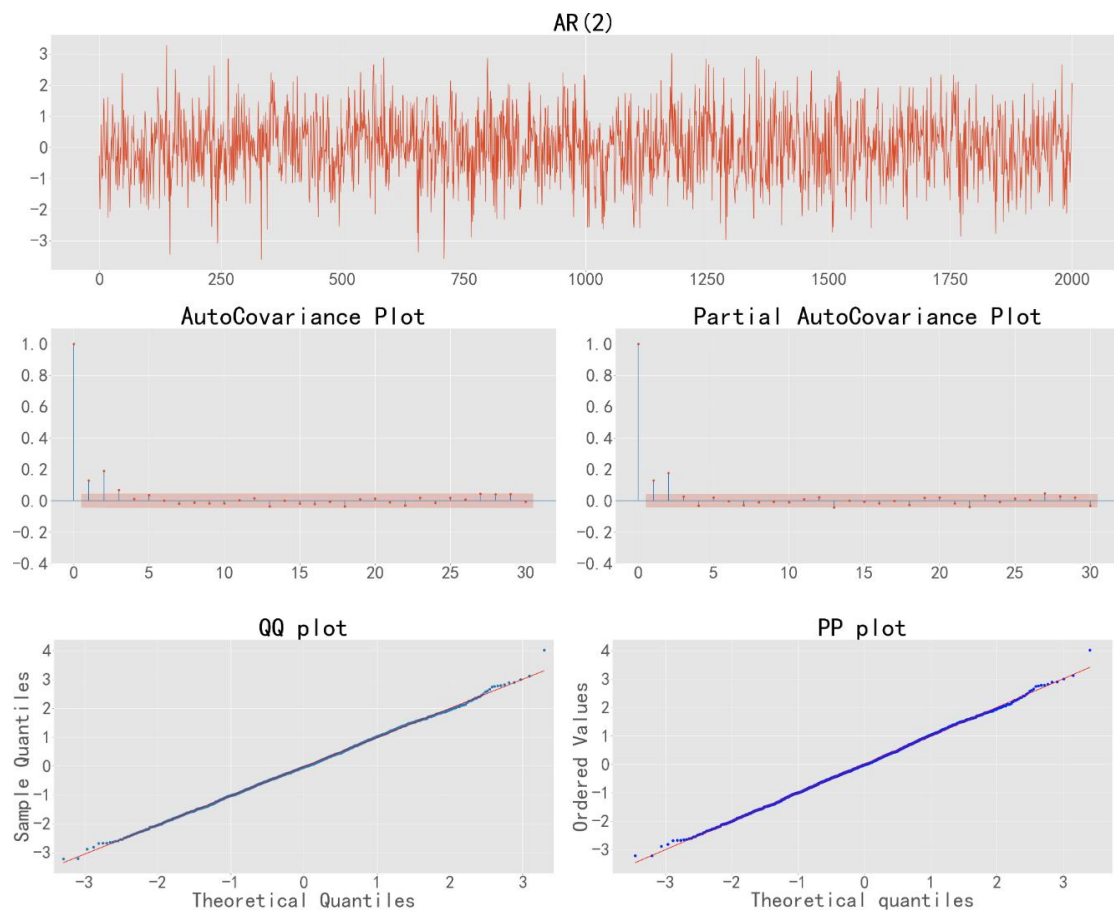
Answer:

AR(1):

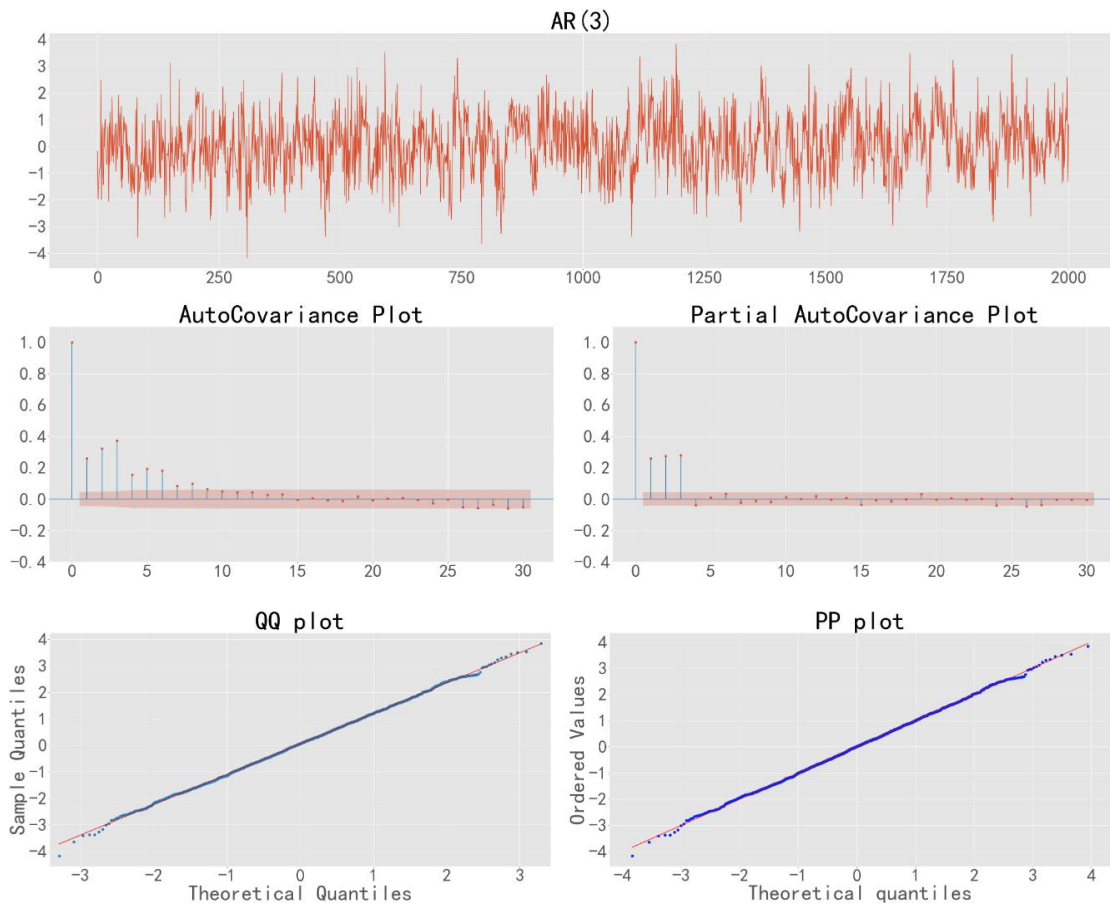




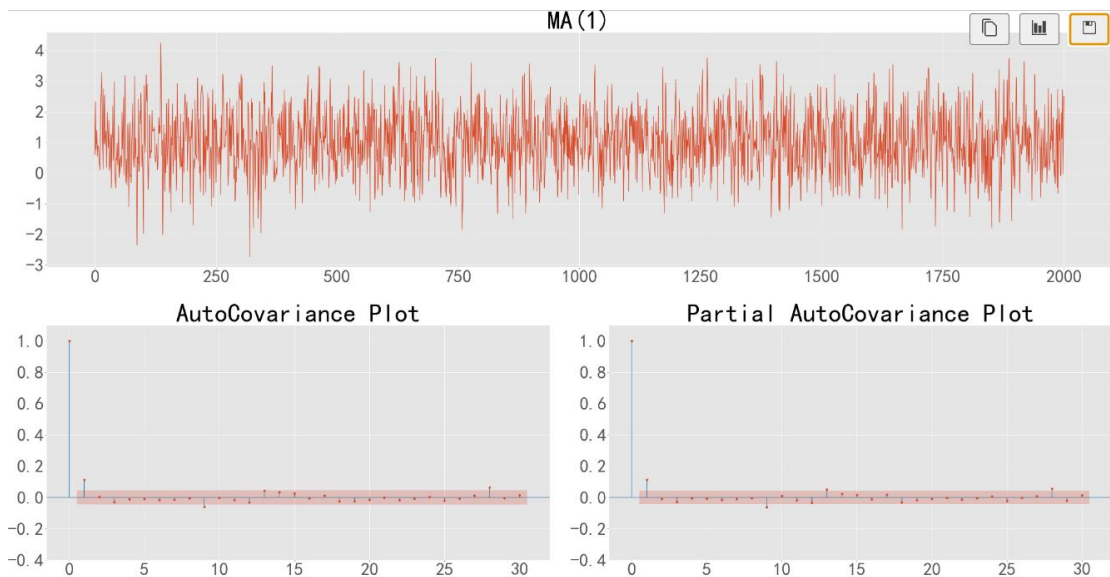
AR(2)

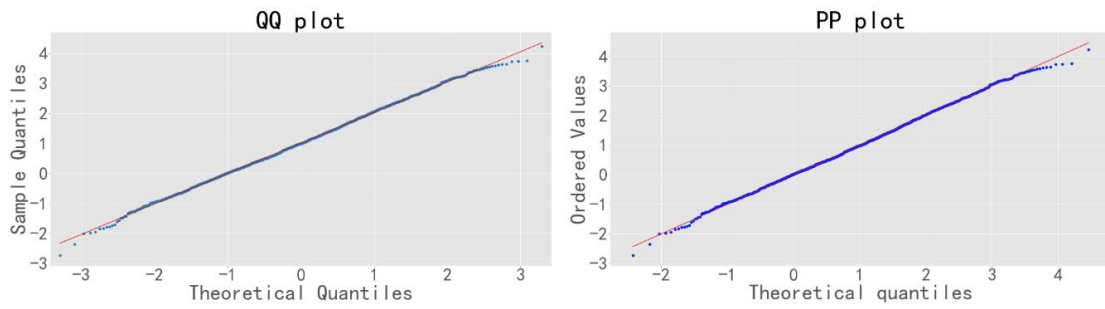


AR(3)

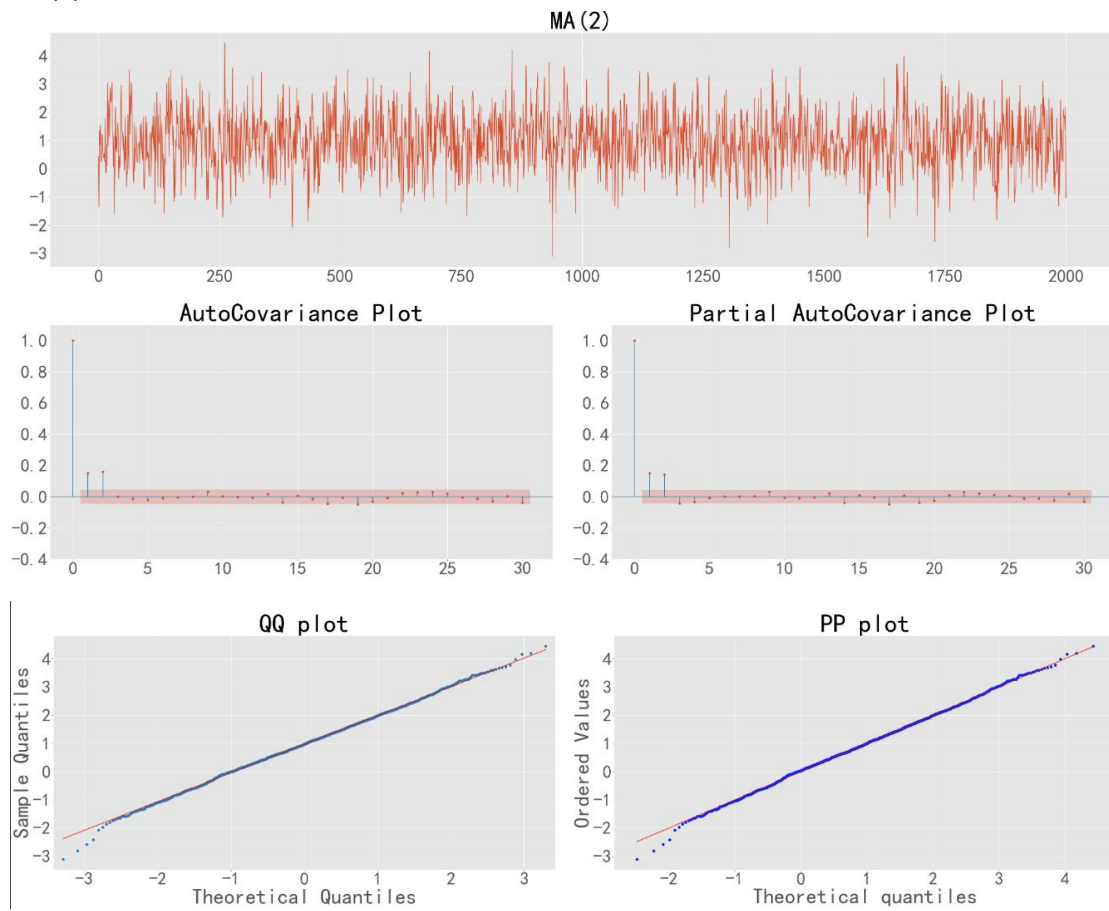


MA(1)

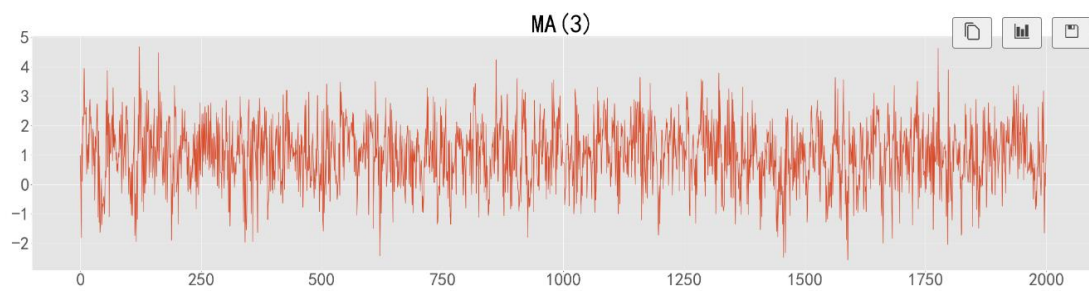


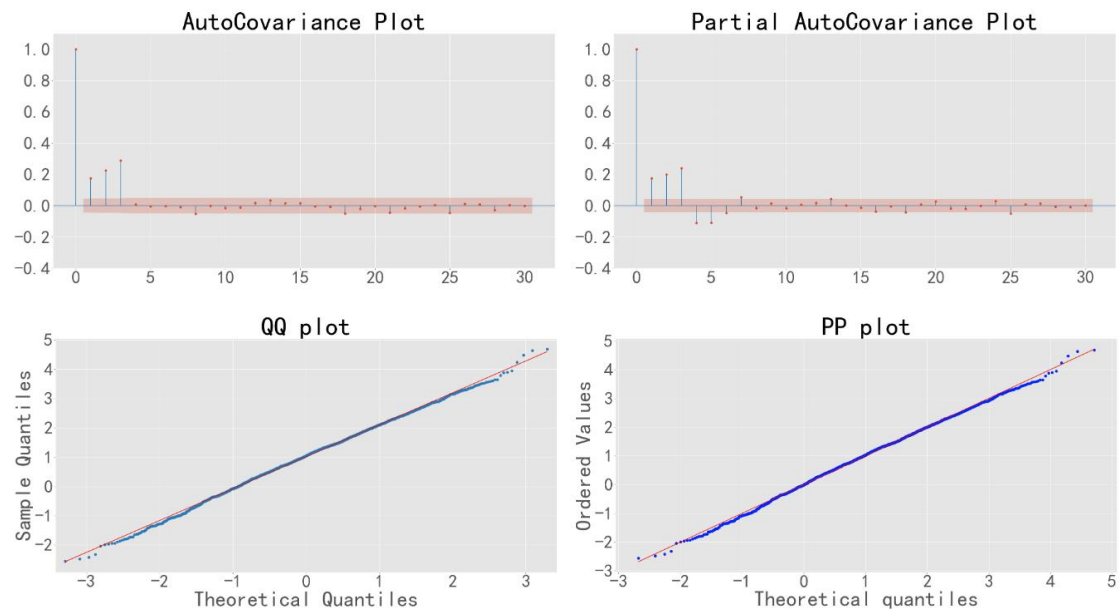


MA(2)



MA(3)





In an AR(p) process, the ACF will have significantly non-zero autocorrelations for the first p lags, and the PACF will have significantly non-zero values for the first p lags and then cut off abruptly. And the ACF will have a slow decay while the PACF will have a sharp cutoff after a few lags.

In an MA(q) process, the ACF will have significantly non-zero autocorrelations for the first q lags, and the PACF will have significantly non-zero values for the first q lag and then cut off abruptly. And the PACF will have a slow decay while the ACF will have a sharp cutoff after a few lags.

In summary, when seeing a slow decay in the ACF but a sharp cutoff in the PACF, it suggests an AR(p) process, whereas when seeing a slow decay in the PACF but a sharp cutoff in the ACF, it suggests an MA(q) process.

So, in an ARMA(p,q) process, the ACF and PACF will both have significantly non-zero values for the first p and q lags, respectively, and then cut off abruptly.