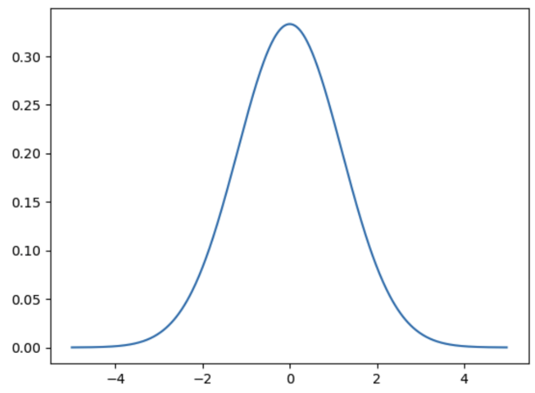
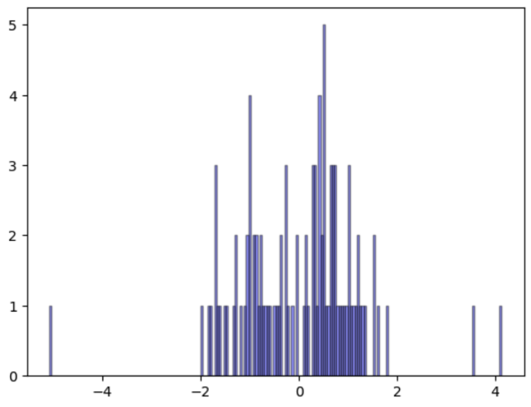
Problem 1:

My function is biased. The given sample's size is 100, which is very small. As we know, when the sample size increases, the bias of variance will get smaller. Since the difference between n and (n-1) will shrink as the sample size gets larger. We know that the variance formula is a factor of the skewness formula, and the skewness formula is a factor of the kurtosis formula, therefore both two functions are biased. Basically, I code those formulas in python. Using a for loop, calculate skewness and kurtosis 2000 times. Using the results I get, to calculate the mean of skewness and kurtosis. Both of their means are not equal to zero. Therefore, it's biased.

Problem 2:

By utilizing the statsmodel package, I calculated const and b1. Which are β0 and β1 respectively in this formula $y = \beta 0 + \beta 1x + \varepsilon$. Just put epsilon to the otherside, we get $\varepsilon = y - \beta 0 - \beta 1x$. By solving this equation, we got a list of ε. I drew two diagrams. The 1st is a normal distribution with mean of epsilon and standard deviation of epsilon. The 2nd is a distplot diagram based on the value of the epsilon list. Graphically, the two graphs are very similar although there is a slight difference, and the data in the middle of the second graph is very dense.





By utilizing the scipy model, I calculated -2log-likelihood values for normal distribution and T distribution, which are -159.99 and -155.47 respectively. The results of beta0 and beta1 of normal distribution fitting into OLS and MLE are equal, which are 0.1198362 0.60520482 respectively.

As we know, under the -2log-likelihood is used to reflect the degree of fit of the model, so the absolute value of the result is better when it's smaller. Therefore, it can fairly say that T distribution is the best fit among those three.

Problem 3:

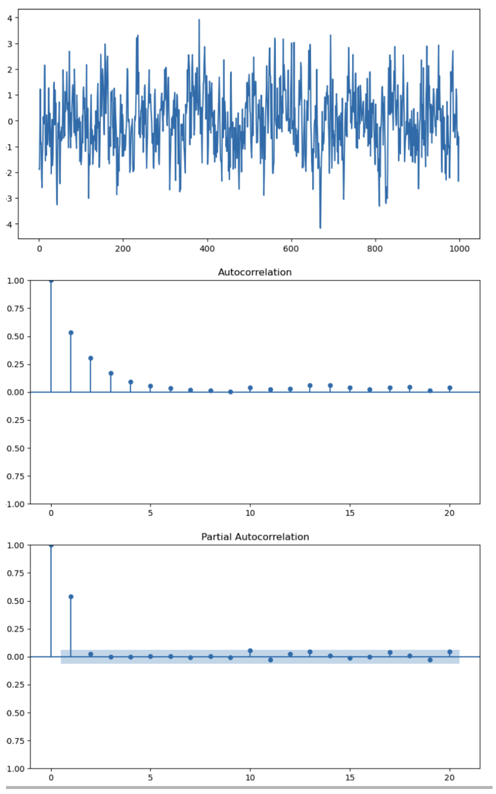
We know that AR model of order 1 is AR(1), which is $R_t = \mu + \phi_1 R_{t-1} + \epsilon_t$, and AR(2) AR(3) is higher order. Since, the parameter's absolute value can't be greater than 1, so I choose 0.6 as parameter. AR(2)'s parameter is 0.3, and AR(3)'s parameter is 0.1. I generate three graphs including ACF and PACF for each order level.

Under the case, ACF has a tail, but PACF cuts tail on some order level. We can see PCAF graphs of AR(1) through AR(3), as the number of x-axis increases, the y-axis value approaches zero. So this series of graphs are fitting into AR.

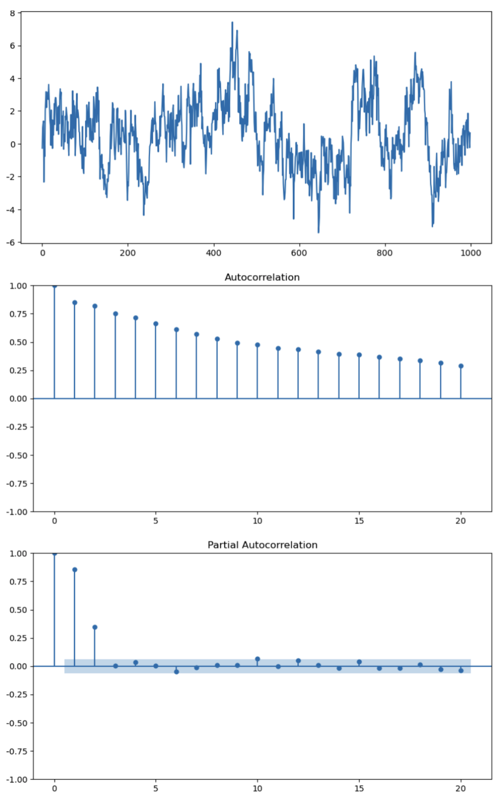
For AR(1), ACF is pretty similar to PCAF, since ACF is equal to PACF under first order. For AR(2) and AR(3), ACF is very different from PCAF.

The PACF is usually non-zero within the lag order of the model and is usually zero outside the order.

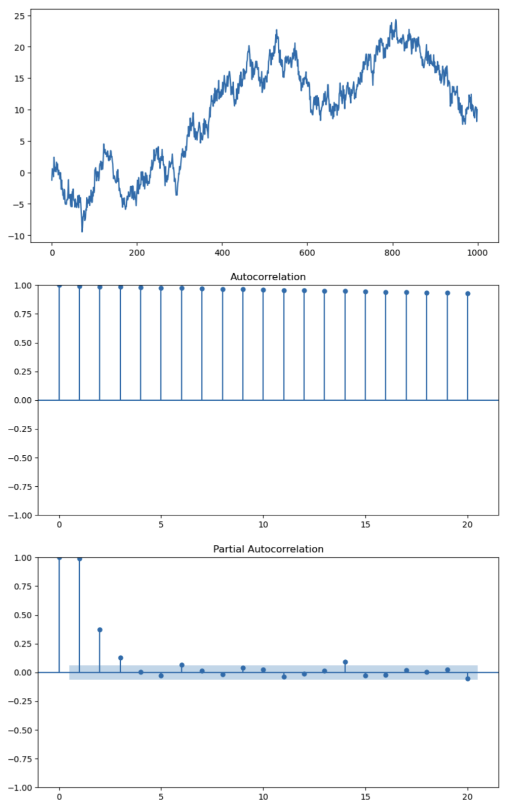
On the other hand, looking at MA’s ACF and PACF graphs, only 1st lag is positive. The rest of the lags are either negative or approach zero.



AR(1)

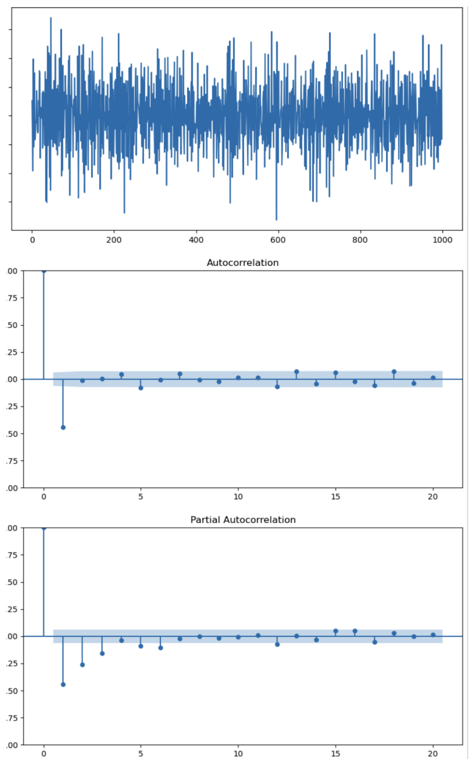


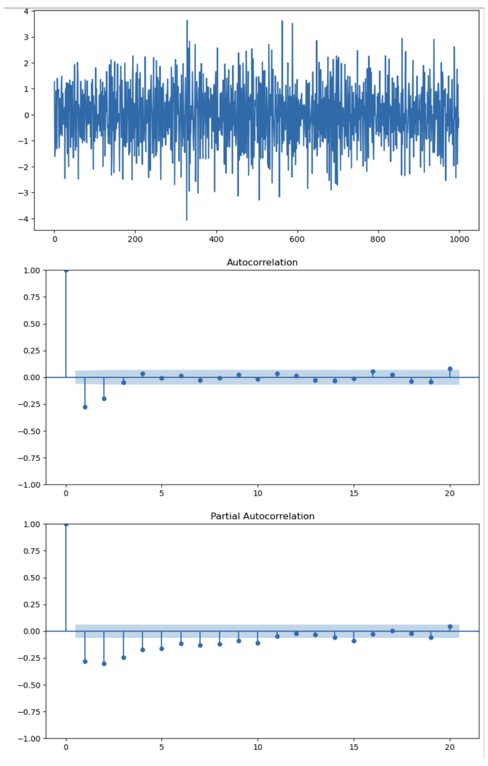
AR(2)



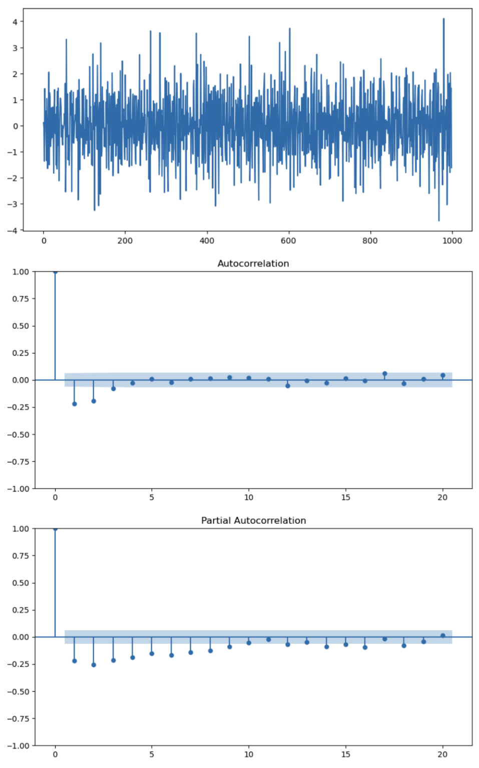
AR(3)

Using the similar way to create graphs for MA(1) through MA(3)





MA(1) MA(2)



MA(3)

Reference:

MLE in python
https://analyticsindiamag.com/maximum-likelihood-estimation-python-guide/

AR model in python
https://goodboychan.github.io/python/datacamp/time_series_analysis/2020/06/08/01-Autoregressive-Models.html

MA model in python
https://goodboychan.github.io/python/datacamp/time_series_analysis/2020/06/08/02-Moving-Average-and-ARMA-Models.html