Problem 1:

Procedure:

First, I set the null hypothesis H0: the function in statistical packages is unbiased, and apply to both skewness and kurtosis, with the significance level of 5%. Then, since I am particularly interested in seeing the effects of different sample sizes when sampling from the standard normal distribution on the result, I follow the steps below to obtain the values of kurtosis and skewness, plus performing the t-test to obtain the conclusion:

1. For different sample sizes 100, 1000, and 10000, randomly sample standard normal r.v.
2. Obtain the skewness and kurtosis given the sample
3. Repeat step 1 and 2 for 1000 times
4. Perform t-test to obtain the t-statistics and corresponding p-values

Implementation:

I wrote a generic function `test_bias` which can accommodate both kurtosis and skewness functions. The first part of the function is about sampling standard normal r.v. and obtaining the kurtosis and skewness, and the second part is performing the t-test using the `ttest_1samp` function from `scipy.stats` and report the results of t-statistics and p-value. One another thing to consider is that the `kurtosis` function from `scipy-stats` calculates the excess kurtosis by default. Thus, when performing the t-test, we should compare the calculated value for kurtosis with the value 0.

Conclusion:

Function/Sample Size	100	1000	10000
Skewness	p-value: 0.15921	p-value: 0.34372	p-value: 0.29924
Kurtosis	p-value: 0.0007	P-value: 0.11496	p-value: 0.69801

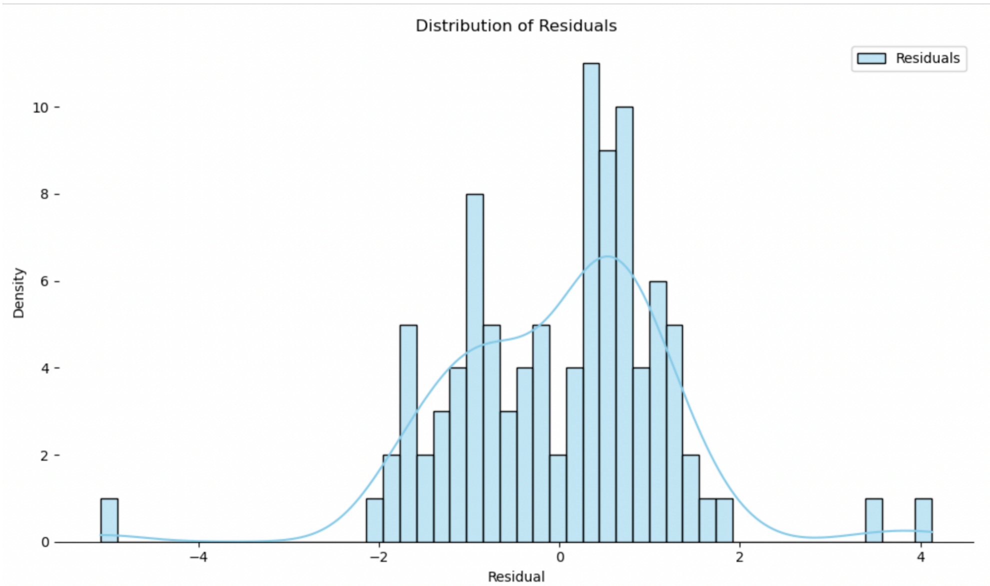
From the summary table above, we saw that only for the sample size of 100, the associated p-value for the kurtosis function is nearly zero. So we should reject the null hypothesis, indicating that the kurtosis function might be biased. For all the other samples sizes with both functions, the associated p-values are greater than the significance level of 5%, suggesting that the skewness function and kurtosis function might not be biased for the sample sizes of 1000 and 10000.

Observations:

1. As the sample size increases, the average skewness and excess kurtosis values are getting closer to their expected values of 0, proving the idea that larger sample sizes lead to better estimation in consideration of the law of large numbers.
2. `scipy.stats` skewness and kurtosis functions seem to produce unbiased results for larger sample sizes.

Problem 2:

OLS:
After reading the data from problem2.csv, I added the intercept column into the design matrix X and fitted an OLS model using statsmodels, and the graph below is the histogram of the error vector (residuals).



We saw that the residuals do show a somewhat bell-shaped distribution, but there are deviations from the standard normal distribution. Next, by doing a Shapiro-Walk test, which is useful in testing whether the residuals are normally distributed.

```
ShapiroResult(statistic=0.9383853673934937, pvalue=0.00015388781321235)
```

Given the low p-value, which is less than the significance level of 5%, we would reject the null hypothesis of the Shapiro-Wilk test, suggesting that the residuals are not normally distributed.

MLE:
To fit the data using MLE under the assumption of normality and then a t distribution of the errors, I first define negative log likelihood functions for both assumptions, and then use the optimization function from `scipy.optimize` to minimize the negative log-likelihood and find the parameters of the model, given an initial guess to the corresponding model parameters. The results are below:
Since a smaller negative log-likelihood indicates a better fit to the data, we saw that the model that assumes a t distribution for the errors is a better fit for the data compared to the model that assumes normally distributed errors.

Also, from the results of AIC and BIC, which are metrics used for model comparison:

```
Log-likelihood using MLE given normality: -159.99209668916234

Log-likelihood using MLE given t error distribution: -155.47297041246662
```

Given that lower AIC and BIC values suggest a better model fit, we saw that both AIC and BIC values are lower for the model that assumes a t distribution for the errors, further confirming that this model is a better fit for the data compared to the model that assumes normally distributed errors.

Besides, we can also obtain the fitted parameters from two models:

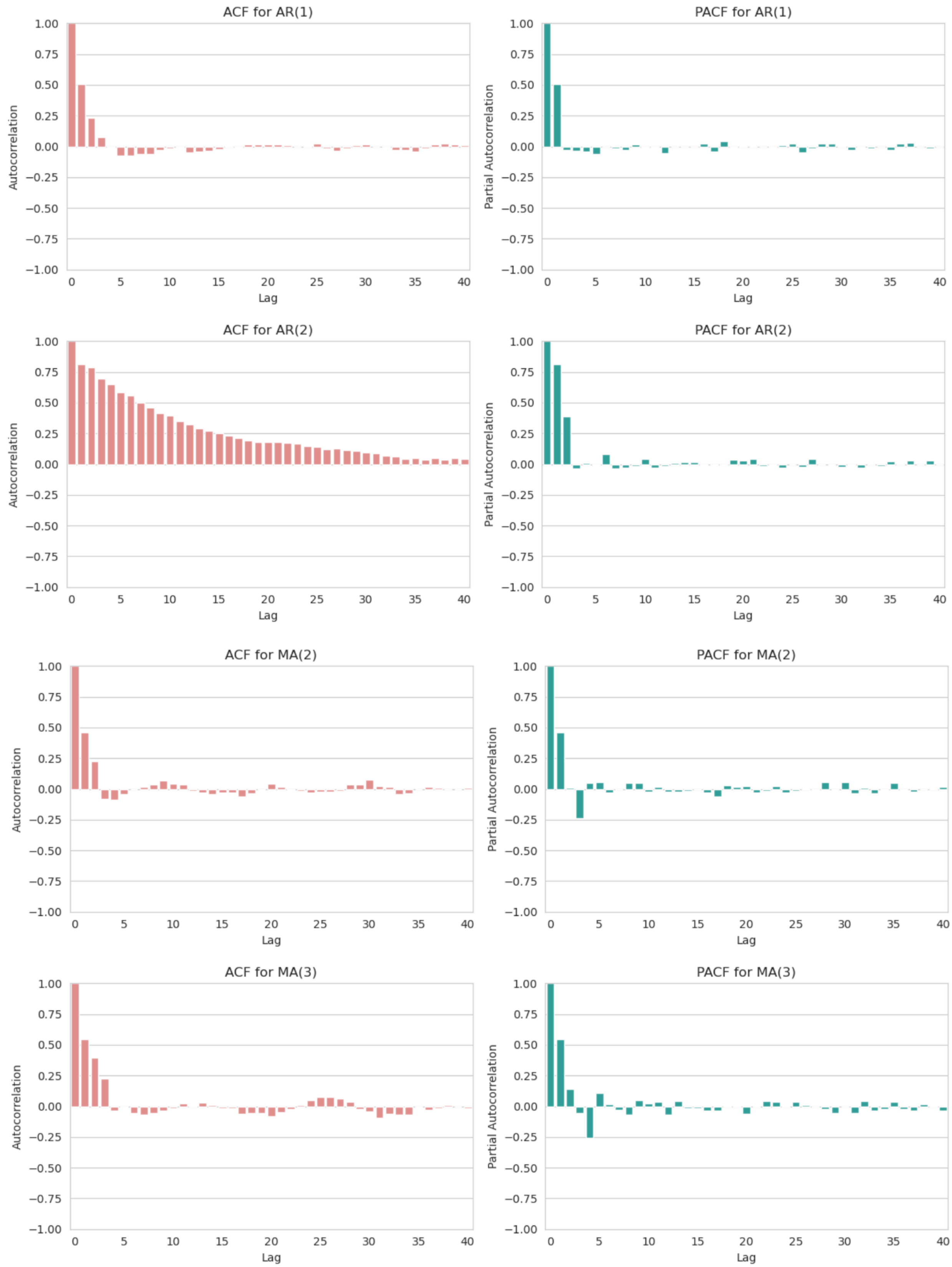
```
Estimated parameters using MLE given normality
Intercept: 0.11983619837146034, slope: 0.6052048204621412, standard deviation: 1.1983941309451727

Estimated parameters using MLE given t error distribution
Intercept: 0.14261409575632913, slope: 0.5575717631662231, standard deviation: 0.9712659596886066, degrees o
f freedom: 6.276561812151229
```

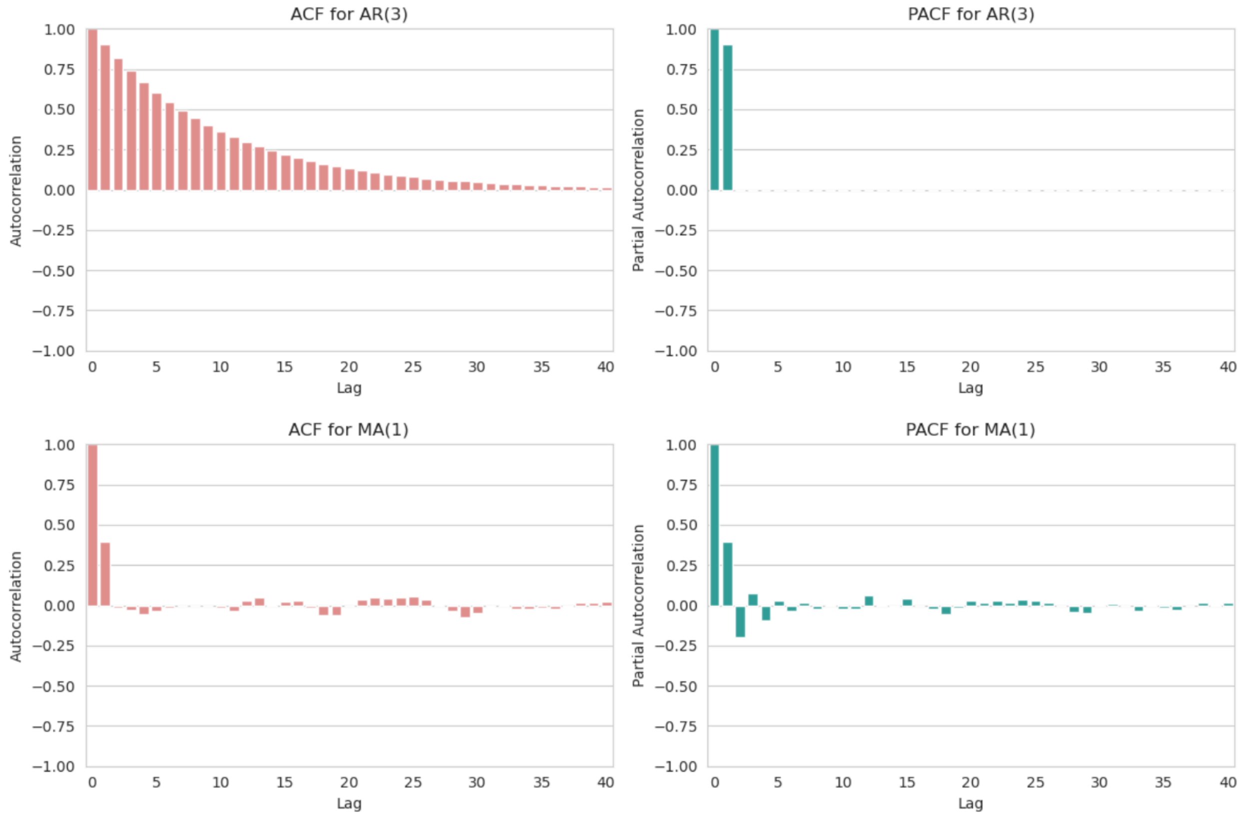
From the results above, we saw that the slope from the t-distribution model is slightly lower than from the normal model. This implies that, under the t-distribution assumption, the relationship between the independent and dependent variables is somewhat weaker. Also, the standard deviation for the t-distribution model is lower than for the normal model. This might suggest that the variability of errors is somewhat tighter when considering t-distributed errors.

Violations of normality in OLS regression can distort hypothesis tests and confidence intervals. If error distribution isn't normal, relying on standard OLS can be deceptive. Using a t-distribution model offers a more adaptive approach, especially when outliers are present or normality is violated.

Problem 3:

ACF and PACF graphs





Upon analyzing the ACF and PACF plots for the AR and MA processes, we can derive the following insights:

<u>AR Processes:</u>

- AR(1): The ACF shows a gradual decline, indicating correlations between the current value and its previous values. In contrast, the PACF reveals a significant correlation only at the first lag, followed by an abrupt cutoff, suggesting an AR(1) model.

- AR(2): The ACF portrays a more extended tail-off, suggesting extended autocorrelations. The PACF, however, has pronounced spikes at the first two lags, confirming the presence of an AR(2) process.

- AR(3): With the ACF still tapering off, the PACF showcases three distinct lags with notable values, implying an AR(3) process.

<u>MA Processes:</u>

- MA(1): The ACF displays a sharp decline after the first lag, reflecting the short-lived correlation. The PACF, on the other hand, exhibits a more gradual decline, indicating potential correlations over an extended range.

- MA(2): The ACF has two noticeable lags, after which it drops, suggesting the impact of the previous two values. The PACF gradually diminishes, indicating a more spread-out partial correlation.

- MA(3): The ACF continues to present three significant lags before a drop, indicating the influence of the past three values. The PACF, however, shows a prolonged tail-off.

Thus, for AR processes, the PACF plot provides a clear indication of the order. The number of pronounced lags in the PACF corresponds to the AR process order. For MA processes, the order is discerned from the ACF plot. The number of significant lags before the ACF drops points to the MA process order.