Problem1

Question:

Skewness and kurtosis functions in statistical packages are often biased. Is your function biased? Prove or disprove your hypothesis.

Solution:

For the first problem, I found that Kurtosis is biased in python and Skewness is more challenging to prove. The function of sample skewness is $S_K = \left[\frac{n}{(n-1)(n-2)}\right] \times \frac{\sum_{i=1}^{n}(x_i-\overline{x})^3}{s^3}$ n is the sample size, and s is the sample standard deviation. The function of kurtosis is $K = \frac{\frac{1}{n}\sum_{i=1}^{n}\left(x_i-\overline{x}\right)^4}{\left(\frac{1}{n}\sum_{i=1}^{n}\left(x_i-\overline{x}\right)^2\right)^2}$ K is for kurtosis. In python, we use the function skew() to calculate skew and the function kurtosis() to calculate kurtosis. Note that the second parameter bias defaults to True. Both equations are biased by default in python, which can be corrected by setting the second parameter to False.

To prove it, I create two empty arrays, "skews" and "kurts" to store the skewness and kurtosis values for each sample. Using the random function, it creates a for loop to generate X samples of 10 random numbers each from a normal distribution. I calculate the skewness and kurtosis for each sample using the skewness and kurtosis functions in Python, respectively. The skewness and kurtosis values are then stored in the corresponding arrays.

Then, I use one-sample t-tests on the skewness and kurtosis arrays to test whether their means are equal to 0. The ttest_1samp function returns a test summary that includes the t-test's outcome and the two-sided p-value. If the p-value is less than a chosen significance level (0.05), we can know the result is a failure to reject or reject the null hypothesis.

Lastly, I choose samples 1000, 100, 50 to print the test summary for the skewness values and kurtosis values. It is clear that the P-value of kurtosis is less than 0.05 and the calculation of kurtosis is biased. The P-value of skewness is more than 0.5 and fails to reject the null hypotheses statistically.

```
Samples 1000

Skews: TtestResult(statistic= -0.27756469409567,pvalue= 0.7814039656243)
```

```
Kurts: TtestResult(statistic= -24.296061990719, pvalue= 7.8921289848e-103)

Samples 100

Skews: TtestResult(statistic= 0.024094472622758, pvalue= 0.9808257512171)

Kurts: TtestResult(statistic= -7.61449152578598, pvalue= 1.5915823018e-11)

Samples 50

Skews: TtestResult(statistic=0.82409084204389, pvalue= 0.413881356828124)

Kurts: TtestResult(statistic= -6.5929698919478, pvalue= 2.83465898133e-08)
```

Problem 2

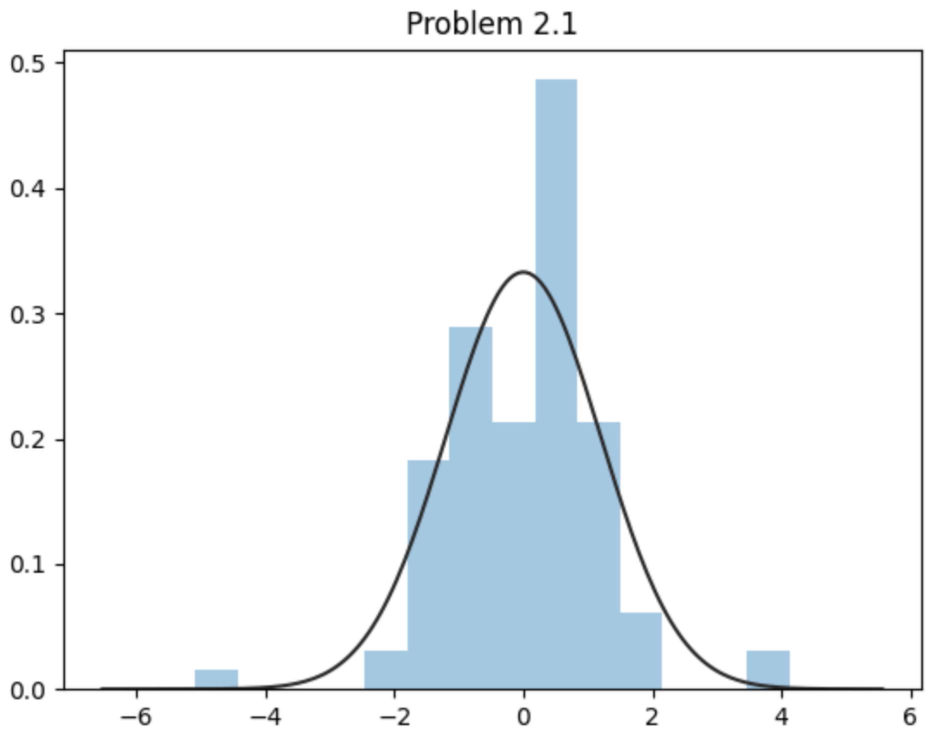
Question:

Q1: Fit the data in problem2.cs using OLS and calculate the error vector. Look at its distribution.How well does it fit the assumption of normally distributed errors?

Solutions:

```
                            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:                        19:16:45   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]
------------------------------------------------------------------------------
const          0.1198      0.121      0.990      0.325      -0.120      0.360
x1             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 normal MLE is used to fit a linear model of the form Y = Xbeta + e where e is normally distributed with mean 0 and variance s^2. The t-MLE is used to fit a linear model of the form Y = Xbeta + e where e is t-distributed with mean 0. First, I calculated the data in problem2.cs by using OLS. To solve the second problem, I calculated the error vector and plotted the result. The error vector did not follow the normal distribution well. Then I calculated Skewness and Kurtosis of “Error”:

Skewness_error: -0.267266585528796

Kurtosis_error: 3.1931010009568785

The kurtosis is 3.2 and “Error” is too kurtotic to be normally distributed.

Q2: 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?

Solution:

I made MLE for the normal and T-distribution models and outputs the estimated parameters, log-likelihood, and Akaike information criterion (AIC) for each model. The AIC measures the relative quality of the models, with a lower AIC indicating a better model. Then I calculated the R-square of these two models, the results are as follows:

```
Normal AIC: 323.9841933783258  BIC: 329.194533750302 Square:0.1946395239189
T-dis  AIC: 314.945940824933   BIC: 320.156281196909 Square:0.1934338097282
```

The results of the R-square are almost the same, and this cannot be the criteria to know which model is the best fit. Noticing that the AIC and BIC of the normal distribution model are higher than T-distribution’s, we can know that the T-distribution is better. We can know that when n gets large, even the smallest deviation from normality will lead to a significant result. As every dataset has some degree of randomness, no single dataset will be a perfectly normally distributed sample. In addition, I find that formal normality tests always reject the huge sample sizes we work with today.

	T-distribution	Normal distribution
AIC	314.9459	323.9842
BIC	320.1562	329.1945
R_square	0.1934338	0.1946395
X	0.5576	0.6052
Intercept	0.1426	0.1198

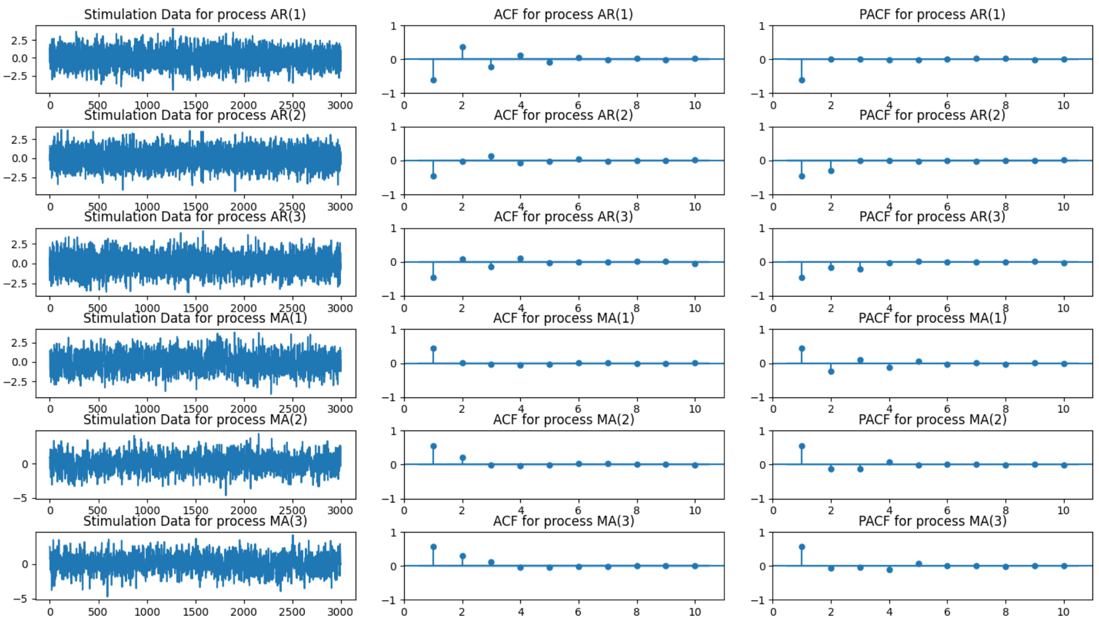
T-distribution has almost the same R_square and lower AIC,BIC,X values compared with Normal distribution.

Problem 3

Question:

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

Solution:



The autocorrelation function (ACF) is a statistical technique that we can use to identify how correlated the values in a time series are with each other. Partial autocorrelation (PACF) is a statistical measure that captures the correlation between two variables after controlling for the effects of other variables. Looking at the graph above, the value of the autocorrelation function (ACF) for an autoregressive (AR) model is always non-zero and it decays as the lag increases. For a moving average (MA) model, the ACF is only non-zero for the number of lags in the model, while the value of the PACF decays as the lag increases. The ACF and PACF provide useful information for identifying the appropriate order of an AR or MA model.