Q1

(a). By using the formula provided in the class note, the mean, variance, skewness, kurtosis are as follow. To take note, all the values provided here are unbiased.

Mean = 1.0490

Variance = 5.4272

Skewness = 0.8819

Kurtosis = 26.1222

Below is the original output from python.

```
calculate_Mean: 1.0489703904839585
calculate_Variance: 5.427220681881727
calculate_Skewness: 0.8819320922598395
calculate_Kurtosis: 26.122200789989723
```

(b) I choose Pandas as the statistical package. By simply states the 4 moments without any modification on Pandas package, below is what I get:

Mean = 1.0490

Variance = 5.4272

Skewness = 0.8833

Kurtosis = 26.2443

Below is the original output from python Pandas.

```
pandas_Mean: 1.0489703904839585
pandas_Variance: 5.427220681881727
pandas_Skewness: 0.8819320922598395
pandas_Kurtosis: 23.2442534696162
```

To take note, by simply stating the kurtosis using pandas package, it would calculates the excessive kurtosis (which is kurtosis – 3). Thus, by adding 3 on the pandas output, we get the comparable kurtosis as the one we calculated in question (a).

(c) Generally speaking, pandas's functions are biased. We verify this by generate a normal distributed random series with mean = 50, variance = 25, skewness = 0 and kurtosis = 3. Than using pandas to calculate 4 moments respectively. Null hypothesis is set as: there is significant difference between theoratical 4 moments and the ones calculated by pandas. By using the t-test for all 4 moments, we find out that except the p_value of mean equals to 0.062, which is larger than 0.05, p_value of other measures are < 0. This indicates that it could reject the null hypothesis and conclude the difference is significant.

Below is what we got from python pandas.

```
Below is the answer for 1c
Calculated Mean: 49.90783139920868
Calculated Variance: 24.384137797474672
Calculated Skewness: 0.02663861269679635
Calculated Kurtosis: 2.9696302462992286
T-test for Mean: Ttest_1sampResult(statistic=-1.866505579853031, pvalue=0.061999912718221216)
T-test for Variance: Ttest_1sampResult(statistic=503.3436722039517, pvalue=0.0)
T-test for Skewness: Ttest_1sampResult(statistic=inf, pvalue=0.0)
T-test for Kurtosis: Ttest_1sampResult(statistic=-inf, pvalue=0.0)
```

Q2

(a). By using the linear regression function embedded in sklearn library, we could get the beta and standard deviation of OLE error as following:

Beta = 0.7753

Standard deviation = 1.0038

Under the assumption of normality, the standard deviation of MLE error could be solved by this function introduced in lecture slide:

$$\widehat{\sigma^2} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2$$

And the result is 1.0038, which equals to the standard deviation of the OLS error. Theoretically, under the assumption of normal distribution of errors, the standard deviation of error of both OLS and MLE should be the same, which is the exact circumstance we observed in here.

Below is the output from python:

```
Below is answer for Q2.a
OLS Coefficients: [0.7752741]
OLS Intercept: -0.08738446427005074
Standard Deviation of the OLS Error: 1.003756319417732
Fitted MLE σ (Standard Deviation of the Error): 1.003756319417732
```

(b). By using the scipy.stats package, we are able to get the log likelihood function for both normal and student-t distribution. By using AIC as the standard to access the goodness of fit, t-distribution is a better fit as its aic_t = 570.5868, which is slightly smaller than the aic_normal = 575.0751. The negative log likelihood of t-distribution is also smaller than that of normal distribution.

Below are all the results generated using scipy.stats

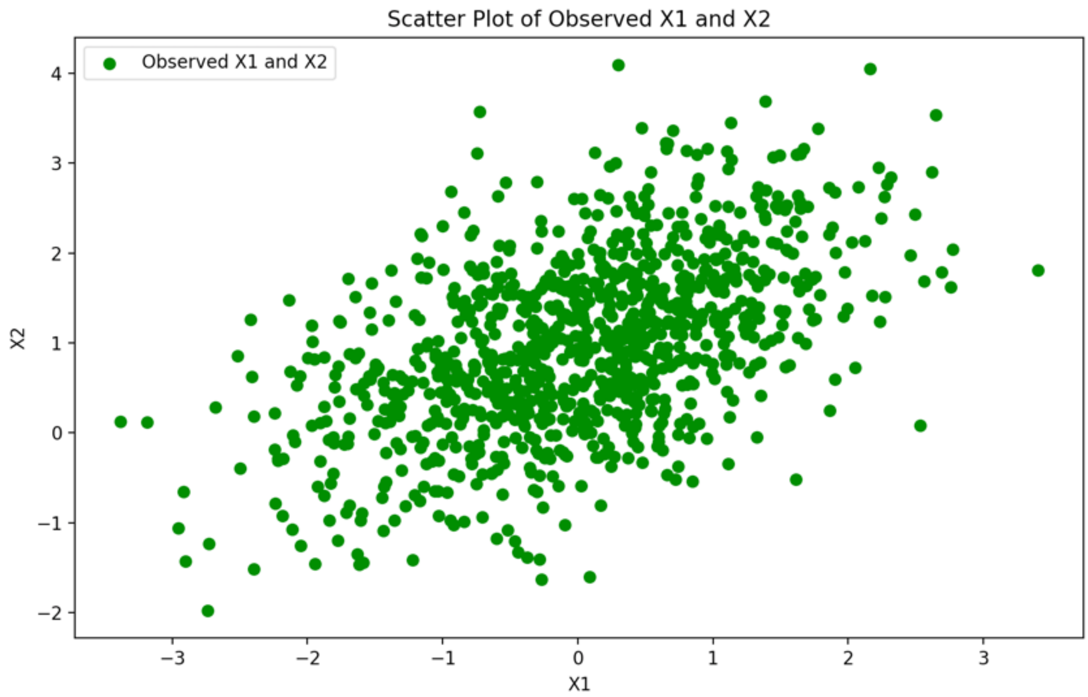
```
T distribution estimated parameters:
Negative Log likelihood of t distribution is: 281.2934031796822
MLE Beta: 0.6750088677808624
MLE Intercept: -0.09726940398925642
DF: 7.159745729538214
Sigma: 0.8551033010978096

Normal distribution estimated parameters:
Negative Log likelihood of normal distribution is: 284.53756305442846
MLE Beta: 0.7752740910719251
MLE Intercept: -0.0873843847772755
Sigma: 1.0037563034041288

AIC for t-distribution models is: 570.5868063593643
AIC for normal distribution models is: 575.0751261088569
```

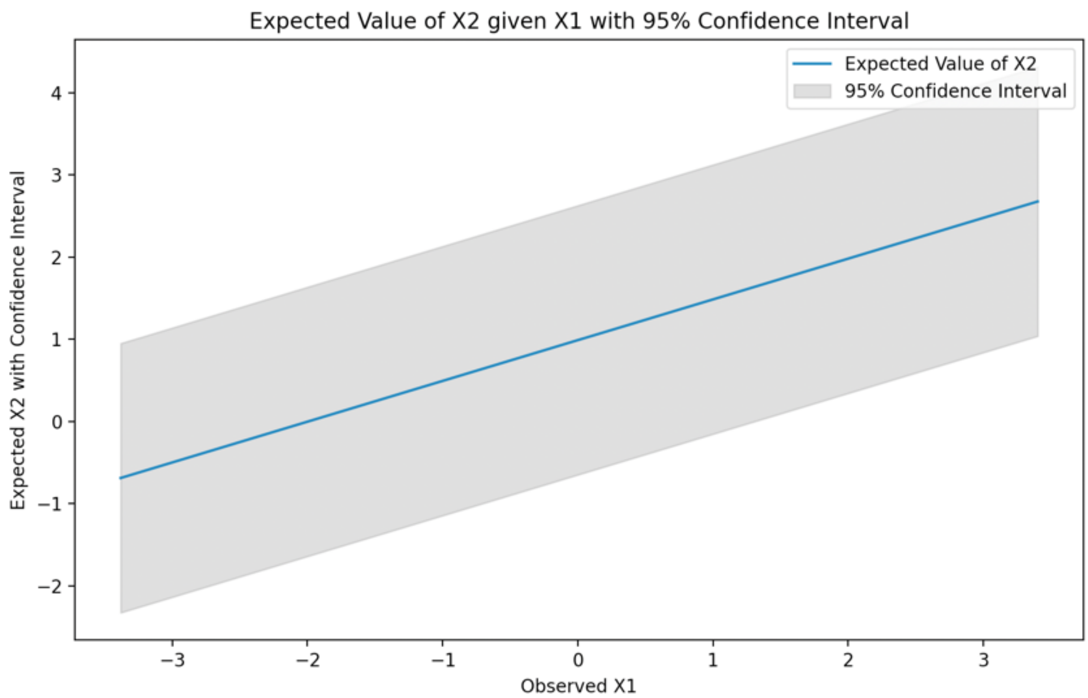
(c). The distribution of x2 should also follow normal distribution, with conditional variance equals to 0.6982 and conditional mean = 0.99 + 0.53/1.07 * (x1-0.001)

Below are the result produced by scipy:

```
mean_x1, mean_x2, cov_xx, cov_yy, cov_xy:  0.0010226951490000004 0.9902438191000001 1.0697746428027168 0.9614732933624849 0.530684554713422
x2_conditional_variance: 0.6982158881639965
```

With the interval being conditional_mean +/- 1.96 * (the square root of conditional_variance), the observed data and expected value along with the 95% confidence interval should be plotted like this:





Q3. AIC is used as the criteria to select the best fit. By iterating through AR(1) to AR(3), as well as from MA(1) to MA(3), AR(3) has the smallest AIC equals to 1436.6598, which indicates AR(3) is the best fit. MA(3) has the smallest AIC among all MA models, and aic_MA(3) = 1536.8677, which is slightly larger than AR(3).

```
The best AR model's order and aic is: 3 1436.6598066945867
The best MA model's order and aic is: 3 1536.8677087350316
AR is a better solution
Best AIC:  1436.6598066945867
Best Order:  3
                               SARIMAX Results
==============================================================================
Dep. Variable:                      x   No. Observations:                  500
Model:                 ARIMA(3, 0, 0)   Log Likelihood                -713.330
Date:                Sat, 27 Jan 2024   AIC                           1436.660
Time:                        00:42:17   BIC                           1457.733
Sample:                             0   HQIC                          1444.929
                                - 500
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const          2.1209      0.085     24.990      0.000       1.955       2.287
ar.L1          0.4515      0.040     11.179      0.000       0.372       0.531
ar.L2         -0.4887      0.037    -13.104      0.000      -0.562      -0.416
ar.L3          0.5047      0.040     12.769      0.000       0.427       0.582
sigma2         1.0132      0.068     14.939      0.000       0.880       1.146
===================================================================================
Ljung-Box (L1) (Q):                   0.02   Jarque-Bera (JB):                 0.84
Prob(Q):                              0.90   Prob(JB):                         0.66
Heteroskedasticity (H):               1.04   Skew:                            -0.03
Prob(H) (two-sided):                  0.81   Kurtosis:                         2.81
===================================================================================
```