Project Week02, by Nurken Abeuov (NETID: na233)

Answer to Problem 1:

1. The first 4 moment values using the normalized formulas in the Week 1 notes:
   Mean: 1.0489703904839585
   Variance: 5.4217934611998455
   Skewness: 0.8806086425277361
   Kurtosis (Excess): 23.122200789989723

2. The first 4 moment values by using the chosen statistical package:
   Mean (package): 1.0489703904839585
   Variance (package): 5.427220681881727
   Skewness (package): 0.8806086425277364
   Kurtosis (Excess, package): 23.122200789989723

3. **Comparison of Results:**

Moment	**Manual from Part 1**	**Statistical package from Part 2**	**Difference**
Mean	1.048970	1.048970	0.000000
Variance	5.421793	5.427221	-0.005428
Skewness	0.880609	0.880609	0.000000
Kurtosis (Excess)	23.122201	23.122201	0.000000

Let's break down the results for each moment:

- **Mean:** The results from both manual and package calculations match perfectly. This indicates that the package's function for calculating the mean is not biased, as the formula for the mean is consistent across both manual and package methods.
- **Variance:** There is a small difference between the manual (biased) and package (unbiased) variance. By default, the statistical package (pandas) calculates the unbiased variance using $n-1$ in the denominator (*ddof=1*), while the manual method divides by n resulting in a biased variance estimate. This demonstrates that the package's variance function is unbiased by default. Additionally, if we modify the code to *x.var(ddof=0)* in the package, it will force the package to divide by n, making the variance calculation biased. This proves that the statistical package can compute both biased or unbiased variance, depending on the value of the *ddof* (degrees of freedom) parameter.
- **Skewness:** By default, the manual and package methods give the same skewness value of 0.880609, but this is because the package's default behavior is to calculate biased skewness *(bias=True)*. Skewness calculated in this way uses the full sample size n, like the manual calculation. To calculate unbiased skewness, the package requires setting *bias=False*, which adjusts the formula to account for sample size, thus correcting for bias. Therefore, the package can calculate both biased and unbiased skewness, depending on how the bias parameter is set.
- **Kurtosis:** The manual and package methods give identical values for kurtosis at 23.122201. Like skewness, kurtosis is biased by default *(bias=True)* in the statistical package, which uses the full sample size n. To obtain unbiased kurtosis, the bias parameter needs to be set to *False*. Again, the package allows the user to compute both biased and unbiased kurtosis by adjusting the bias parameter.

Project Week02, by Nurken Abeuov (NETID: na233)

Answer to Problem 2:

1. **OLS Results:**
   Intercept (Beta 0): -0.0874
   Slope (Beta 1): 0.7753
   Standard Deviation of Residuals (OLS): 1.0038

   MLE Results:
   Intercept (Beta 0): -0.0874
   Slope (Beta 1): 0.7753
   MLE Estimate of σ: 1.0038

   Comparison Between OLS and MLE: The beta values (intercept and slope) for both OLS and MLE are nearly identical. The standard deviation of residuals (OLS) is 1.0038, which is also nearly identical to the fitted MLE σ of 1.0038.

   Conclusion: The results show no significant difference between the OLS and MLE estimates. This is expected because under the assumption of normality, the OLS and MLE estimators for a linear regression model with constant variance yield the same parameter estimates for the coefficients and error variance.
   Therefore, the differences are minimal, and both methods provide consistent results in this case.

2. **Model Comparison Results:**
   MLE Normal Intercept (Beta 0): -0.0874
   MLE Normal Slope (Beta 1): 0.7753
   MLE Normal Sigma: 1.0038

   MLE T-distribution Intercept (Beta 0): -0.0918
   MLE T-distribution Slope (Beta 1): 0.6154
   MLE T-distribution Sigma: 0.7414

   After receiving the above outputs of the comparison of the fitted parameters for the MLE under the normality assumption and MLE under the t-distribution assumption. I can now observe how the parameters differ between these models. To assess the best fit, I would look at goodness-of-fit measures such as log-likelihood values, AIC, and BIC. The t-distribution model may offer a better fit in cases where the errors have heavier tails, while the normality assumption might be sufficient if the residuals are more normally distributed.
   Let's calculate the log-likelihoods and AIC/BIC values for both the normality assumption and t-distribution assumption models with the following functions: calculate_aic, log_likelihood_normal, log_likelihood_t, and calculate_bic.

   Output:
   Log-Likelihood, AIC, And BIC Comparison:

Normal Distribution Model:
Log-Likelihood: -284.5376
AIC: 575.0751
BIC: 584.9701

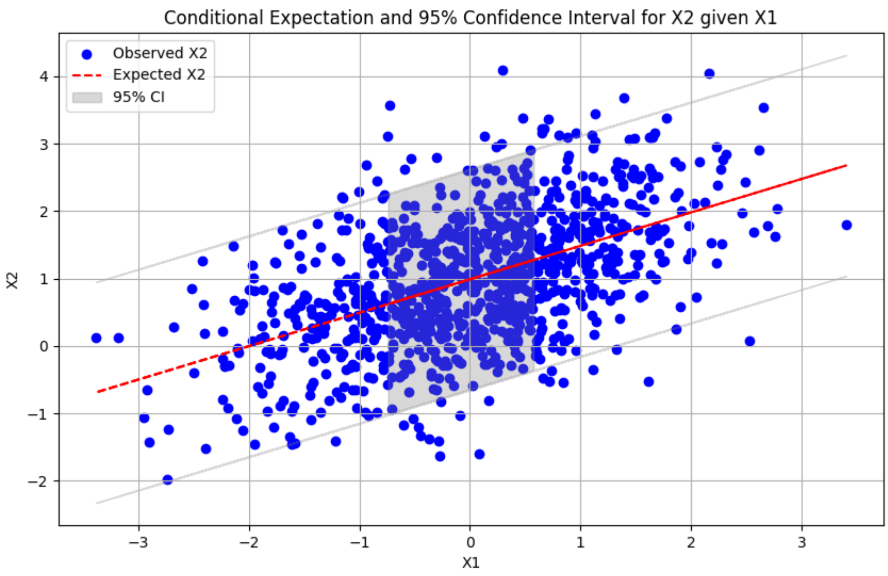
T-Distribution Model:
Log-Likelihood: -284.0546
AIC: 574.1092
BIC: 584.0042

Conclusion:
Log-Likelihood: The t-distribution model has a higher log-likelihood (-284.0546) compared to the normal distribution model (-284.5376). It means that the t-distribution model fits the data better because it provides a higher likelihood of observing the actual data under this model's assumptions.
AIC: The t-distribution model has a lower AIC (574.1092) than the normal distribution model (575.0751). It explains that the t-distribution not only fits the data better but also achieves this fit with reasonable complexity.
BIC: The t-distribution model has a slightly lower BIC (584.0042) than the normal distribution model (584.9701). This confirms that the t-distribution model is a better fit overall, even when penalizing complexity more strongly.

While the differences in log-likelihood, AIC, and BIC are small, all of them consistently favor the t-distribution model. This indicates that the t-distribution fits the data better, likely due to the data having heavier-tailed residuals.

3. The plot below shows the observed values of X_2, the expected conditional values of X_2 given X_1, and the 95% confidence intervals. The red dashed line represents the expected values, the blue points are the observed values, and the shaded gray area represents the 95% confidence intervals.

Project Week02, by Nurken Abeuov (NETID: na233)



4. In the context of the multiple linear regression model:
   $Y = X\beta + \epsilon$, where Y is the $n \times 1$ vector of responses, X is the $n \times p$ matrix of predictors, β is the $p \times 1$ vector of regression coefficients, and ϵ is the error term, which is assumed to follow a normal distribution $\epsilon \sim N(0, \sigma^2)$.

   Step 1: Likelihood function
   The likelihood function is the joint probability density of the observed data Y, conditional on X, β, and σ^2. Since the errors ϵ are normally distributed, the distribution of Y is also multivariate normal:
   $Y \sim N(X\beta, \sigma^2 I)$

   Therefore, the probability density function for Y is:

   $f(Y|X, \beta, \sigma^2) = 1/(2\pi\sigma^2)^{n/2} \exp(-1/2\sigma^2 \, (Y-X\beta)^T \, (Y-X\beta))$

   The likelihood function is the product of these densities for all observations (or equivalently, for the vector Y):

   $L(\beta, \sigma 2|Y, X) = 1/(2\pi\sigma^2)^{n/2} \exp(-1/2\sigma^2 \, (Y-X\beta)^T \, (Y-X\beta))$

   Step 2: Log-likelihood function
   The log-likelihood function is more convenient to work with, and it is the natural logarithm of the likelihood function:
   $\log L(\beta, \sigma^2|Y, X) = -n/2 \log(2\pi) - n/2 \log(\sigma^2) - 1/2\sigma^2 \, (Y-X\beta)^T \, (Y-X\beta)$

   Step 3: Maximizing with respect to β
   To find the MLE of β, we first take the partial derivative of the log-likelihood function with respect to β and set it equal to zero:

$\partial/\partial\beta \log L(\beta, \sigma^2|Y, X) = 1/\sigma^2 X^T (Y-X\beta) = 0$

Solving for β:

$X^TY - X^TX\beta = 0$

$\beta = (X^TX)^{-1} X^TY\beta=0$

Therefore, the MLE of β is:

$\beta=(X^TX)^{-1} X^TY$

Step 4: Maximizing with respect to σ^2

Now, we take the partial derivative of the log-likelihood function with respect to σ^2 and set it equal to zero:

$\partial/\partial\sigma^2 \log L(\beta, \sigma^2|Y, X) = -n/2\sigma^2 + 1/2\sigma^4 (Y-X\beta)^T (Y-X\beta) = 0$

Solving for σ^2:

$n/2\sigma^2 = 1/2\sigma^4 (Y-X\beta)^T (Y-X\beta)$

$\sigma^2 = 1/n (Y-X\beta)^T (Y-X\beta)$

Substituting β into this expression:

$\sigma^2 = 1/n (Y-X\beta)^T (Y-X\beta)$

Since $\beta=(X^TX)^{-1}X^TY$, we can express the residual sum of squares as:

$\sigma^2 = 1/n (Y^TY-Y^TX (X^TX)^{-1} X^TY)$

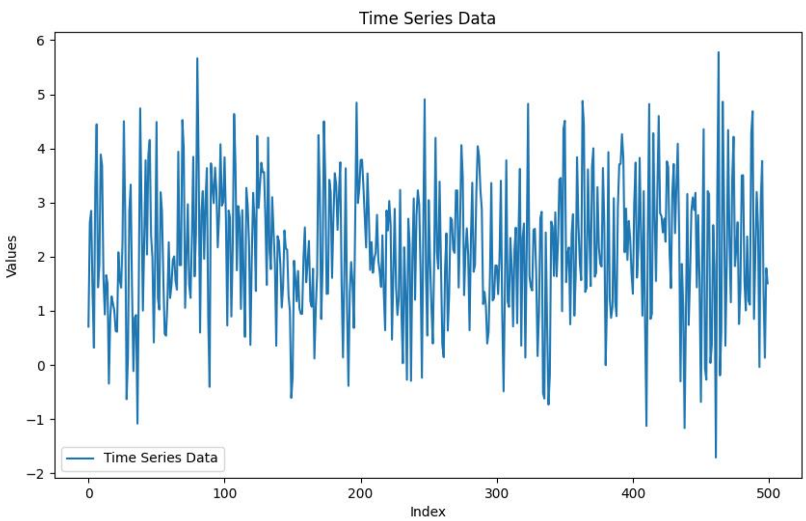
Final Results:

The MLE of β is: $\beta = (X^TX)^{-1} X^TY$

The MLE of σ^2 is: $\sigma^2 = 1/n (Y-X\beta)^T (Y-X\beta)$

<u>Answer to Problem 3:</u>

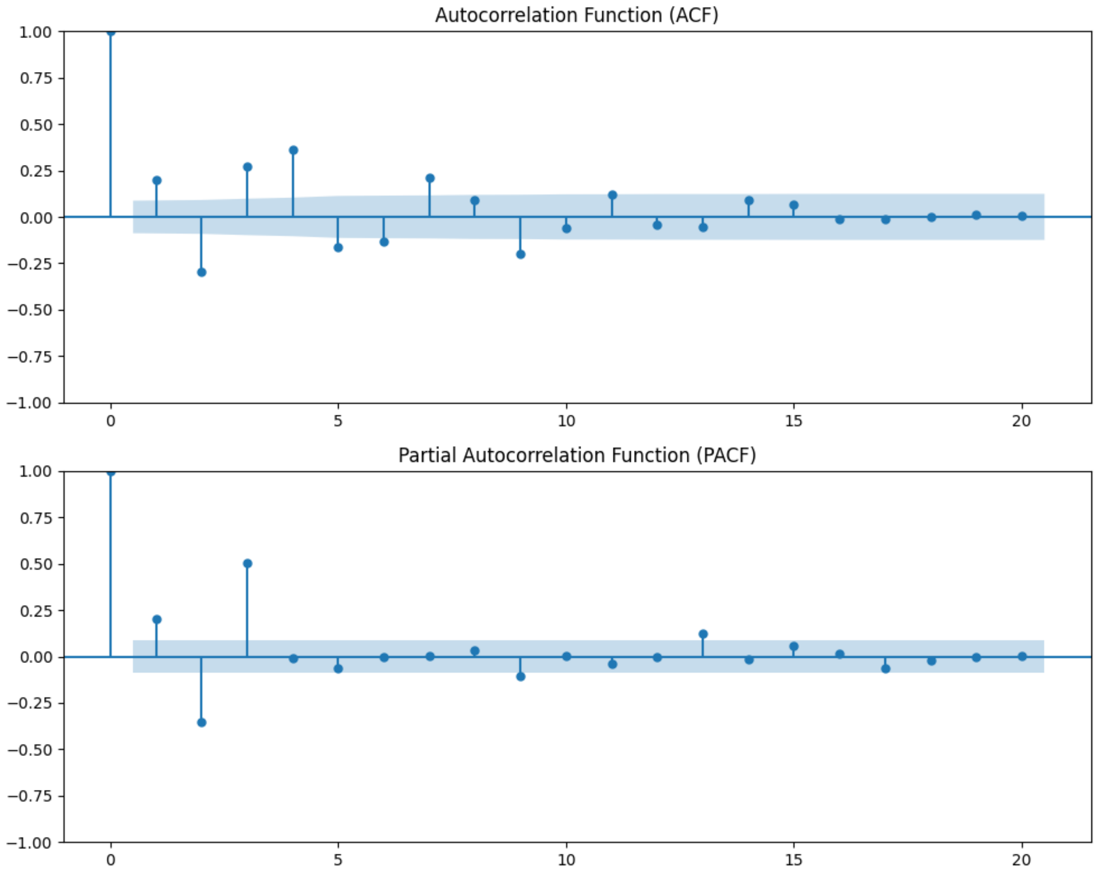
This time series plot provides a visual representation of our data.



Project Week02, by Nurken Abeuov (NETID: na233)

And, now to determine the best AR(n) or MA(n) model for the data, I will examine the autocorrelation function (ACF) and partial autocorrelation function (PACF) plots. These plots should help us to indicate whether an AR or MA process is more appropriate and suggest the appropriate lag order (n).

Here are the generated ACF and PACF plots.



Our ACF plot shows significant spikes at lag 1 and gradually decays, suggesting that a moving average (MA) model might be appropriate.

Also, our PACF plot shows significant spikes at lag 1 and possibly lag 2, indicating that an autoregressive (AR) process could also be relevant.

Based on these observations, AR(1), AR(2), MA(1), and MA(2) could potentially be good models for our data. I'll now fit AR(1), AR(2), AR(3), and MA(1), MA(2), MA(3) models and compare their performance to determine the best fit based AIC and BIC criteria.

The output of models:

Sorted ARIMA Model Results:

	Model	AIC	BIC
0	AR(3)	1436.659807	1457.732847
1	MA(3)	1536.867709	1557.940749
2	MA(2)	1537.941206	1554.799639
3	MA(1)	1567.403626	1580.047451
4	AR(2)	1581.079266	1597.937698
5	AR(1)	1644.655505	1657.299329

Project Week02, by Nurken Abeuov (NETID: na233)

Conclusion: Based on the AIC criterion, the AR (3) model has the lowest AIC, indicating it is the best fit for the data among the AR and MA models.

It suggests that a higher-order AR process (AR(3)) fits the data better than the MA models, which confirms the partial autocorrelation pattern seen in the PACF plot, where significant spikes were observed at higher lags.

Therefore, the AR(3) model is the best fit, confirming our hypothesis based on the ACF and PACF analysis.