```
% First step is to import our given data
Age = [3, 4, 5, 6, 7, 8, 9, 11, 12, 14, 15, 16, 17];
WingLength = [1.4, 1.5, 2.2, 2.4, 3.1, 3.2, 3.2, 3.9, 4.1, 4.7, 4.5, 5.2, 5.0];

% The next step is to fit a linear model to the data
X = [ones(length(Age), 1) Age.'];
b = X\WingLength.';

% This is code to extract the slope and intercept values of the linear
% regression model
slope = b(2);
intercept = b(1);

% This is code to visualize the equation for the fitted linear regression
% line
fprintf('Linear Regression Line: y = %.2fx + %.2f\n', slope, intercept);
```

```
Linear Regression Line: y = 0.26x + 0.83
```

```
% The final step is to then plot the data points as a scatter plot as well
% as the linear regression line
scatter(Age, WingLength, 'b');
hold on;
plot(Age, X*b, 'r');
xlabel('Age');
ylabel('Wing Length');
title('Linear Regression plot comparing Age and Wing Length');
legend('Data Points', 'Linear Fit');
hold off;

% This is code to specify the position of the figure legend
legend("Position", [0.71793,0.33253,0.17054,0.068475])
```


```
% Switching gears to looking at the null hypothesis - in order to see
% whether or not we can reject the null hypothesis we want to see if the
% slope of the line is significantly different from 0 or not - a slope of 0
% would indicate that there is no relationship between the two variables in
% question

% To first look at this we need to extract the p-value related to the slope
% but to do this ChatGPT wanted me to use different code to re-create the
% linear model

% Using the fitlm function to recreate my linear regression model
linearmodel = fitlm(Age,WingLength)
```

```
linearmodel =
Linear regression model:
    y ~ 1 + x1
```

```
Estimated Coefficients:
                   Estimate       SE        tStat       pValue  
                   ________    ________    ______    __________

    (Intercept)    0.82962      0.16774    4.9459    0.00043849
    x1             0.26468     0.015559    17.012    3.0097e-09


Number of observations: 13, Error degrees of freedom: 11
Root Mean Squared Error: 0.256
R-squared: 0.963,  Adjusted R-Squared: 0.96
F-statistic vs. constant model: 289, p-value = 3.01e-09
```

```
% Extracting the p-value for the slope
pValue_slope = linearmodel.Coefficients.pValue(2)
```

```
pValue_slope = 3.0097e-09
```

```
% Setting my significance level at 0.05
alpha = 0.05
```

```
alpha = 0.0500
```

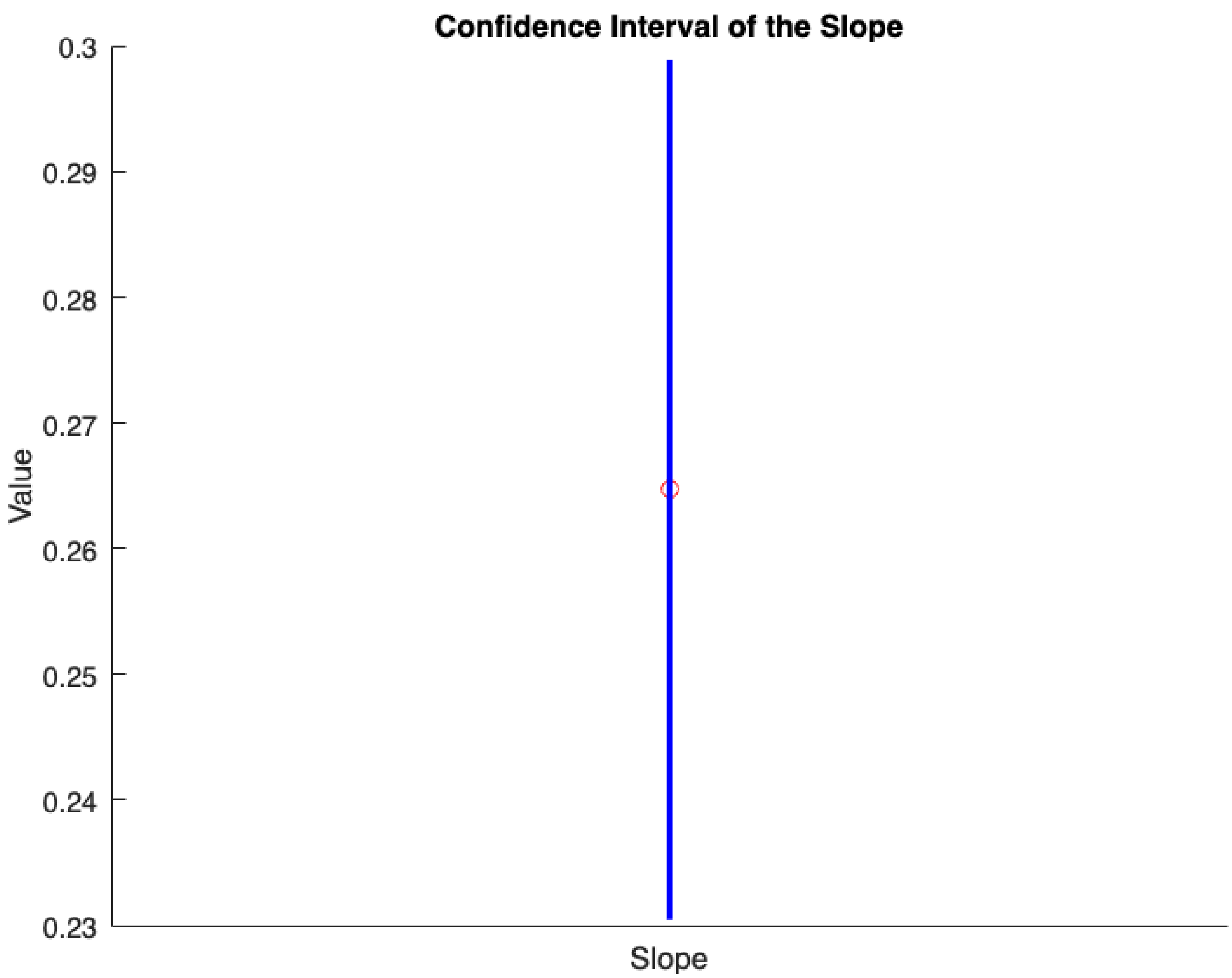
```
% Testing the null hypohtesis by seeing how my calculated p-value relates
% to the significance level I set
if pValue_slope < alpha
    fprintf('Reject the null hypothesis. There is a significant relationship between Ag
else
    fprintf('Do not reject the null hypothesis. There is no significant relationship be
end
```

```
Reject the null hypothesis. There is a significant relationship between Age and WingLength.
```

```
pValue_slope = linearmodel.Coefficients.pValue(2);

% Based on the code I just ran I was able to successfully reject the null
% hypothesis and conclude that there is in fact a significant relationship
% between age and wing length
```

```
% I am now transitioning to adding confidence intervals to my slope
% For this I will continue to use the fitlm function from the previous
% section

% Code to calculate confidence Intervals for the parameters from my linear
% regression
confIntervals = coefCI(linearmodel);
slopeConfInterval = confIntervals(2, :);

% Code to create new figure that includes Slope Confidence Intervals
figure;
```

```
hold on;

% Plotting a point representing the estimated slope
plot(1, linearmodel.Coefficients.Estimate(2), 'ro');

% Adding a vertical line representing the confidence interval of the slope
line([1 1], slopeConfInterval, 'Color', 'b', 'LineWidth', 2);

% Adding Labels and Title
xlabel('Slope');
ylabel('Value');
title('Confidence Interval of the Slope');
xlim([0.5 1.5]); % To better visualize the vertical line
set(gca,'XTick',[]);
hold off;
```




```

```

```
% Calculating the R-squared value - this was actually already done for me
```

```
% using the fitlm function!!
linearmodel
```

```
linearmodel = 
Linear regression model:
    y ~ 1 + x1

Estimated Coefficients:
                   Estimate       SE        tStat       pValue  
                   ________    ________    ______    __________

    (Intercept)    0.82962      0.16774    4.9459    0.00043849
    x1             0.26468     0.015559    17.012    3.0097e-09


Number of observations: 13, Error degrees of freedom: 11
Root Mean Squared Error: 0.256
R-squared: 0.963,  Adjusted R-Squared: 0.96
F-statistic vs. constant model: 289, p-value = 3.01e-09
```

```
% Therefore, based on this function the R-squared value of this linear
% regression is 0.963
```

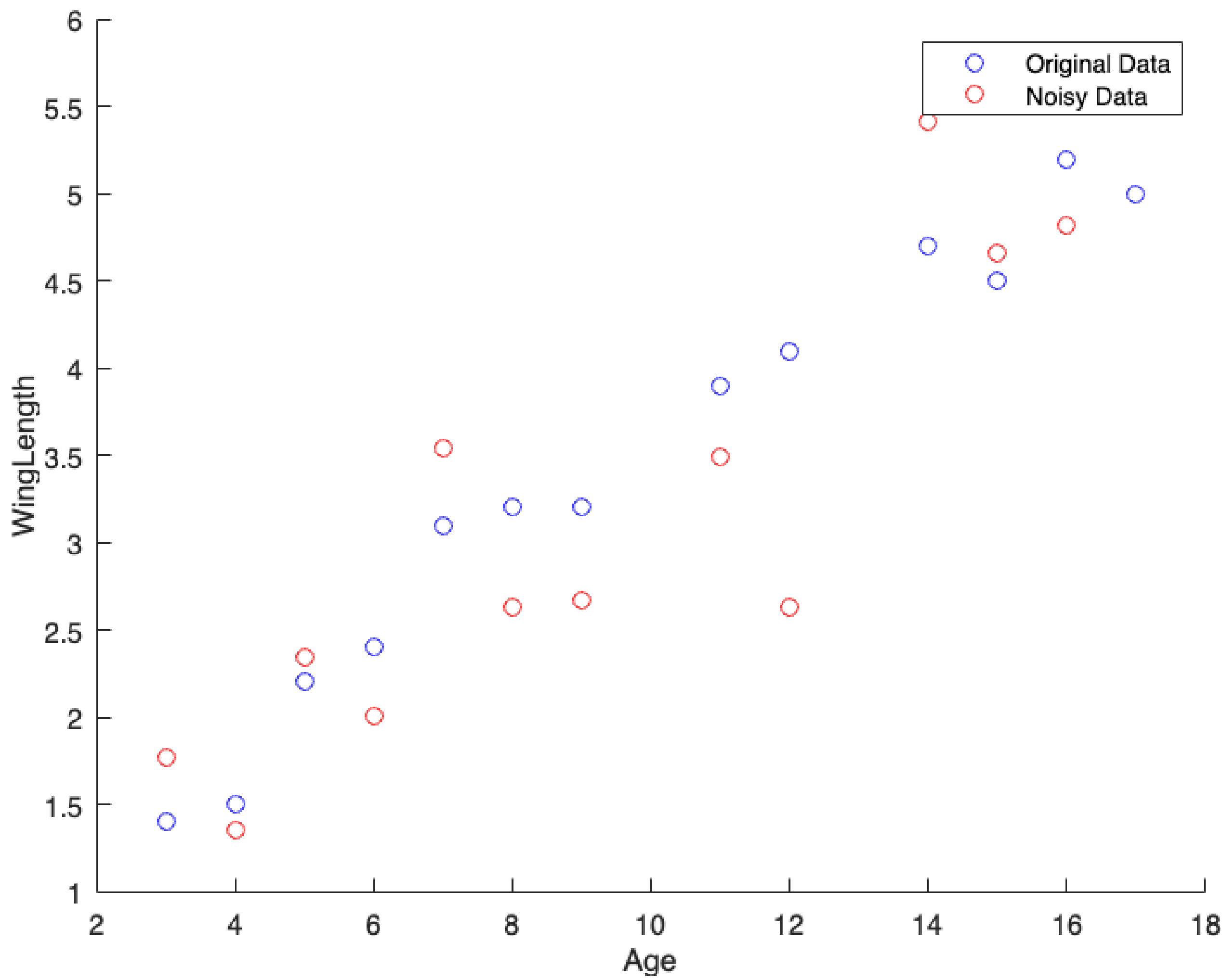
```
% Switching gears to trying to calculate Pearson's r

% First step
R = corrcoef(Age, WingLength);

% Second step
pearsons_r = R(1, 2);

% Display the result
fprintf("Pearson's correlation coefficient (r) between Age and WingLength is %.2f\n", p
```

```
Pearson's correlation coefficient (r) between Age and WingLength is 0.98
```

```
% Based on this generated code Pearson's r was calculated to be 0.98
```

```
% The last step is to add noise to the data and then see how that changes
% the linear regression

% Define the standard deviation of the noise - this is for us to define a
% given parameter of the noise
noise_std_dev = 0.5;

% This is code to generate and add noise to the data
noise = noise_std_dev * randn(1, length(WingLength));
noisyWingLength = WingLength + noise;
```

```
% This is code to now plot our original data vs our noisy data
figure;
scatter(Age, WingLength, 'b'); % Original data in blue
hold on;
scatter(Age, noisyWingLength, 'r'); % Noisy data in red
xlabel('Age');
ylabel('WingLength');
legend('Original Data', 'Noisy Data');
hold off;
```




```
% The final step I will do is compare the linear regression models to one
% another - below is the linear regression model for our original data
linearmodel
```

```
linearmodel =
Linear regression model:
    y ~ 1 + x1

Estimated Coefficients:
```

```
                   Estimate       SE        tStat       pValue  
                   ________    ________    ______    __________

    (Intercept)    0.82962      0.16774    4.9459    0.00043849
    x1             0.26468     0.015559    17.012    3.0097e-09


Number of observations: 13, Error degrees of freedom: 11
Root Mean Squared Error: 0.256
R-squared: 0.963,  Adjusted R-Squared: 0.96
F-statistic vs. constant model: 289, p-value = 3.01e-09
```

```
% This is the new linear regression model for our noisy data
linearmodelnoisy = fitlm(Age,noisyWingLength)
```

```
linearmodelnoisy = 
Linear regression model:
    y ~ 1 + x1

Estimated Coefficients:
                   Estimate       SE        tStat       pValue  
                   ________    ________    ______    __________

    (Intercept)    0.64684      0.41672    1.5522       0.14889
    x1             0.27248     0.038653    7.0493    2.1284e-05


Number of observations: 13, Error degrees of freedom: 11
Root Mean Squared Error: 0.635
R-squared: 0.819,  Adjusted R-Squared: 0.802
F-statistic vs. constant model: 49.7, p-value = 2.13e-05
```

```
% One can see that adding in noise not only changes out intercept and slope
% but also changes the R-squared value and a lot of other important
% parameters that are used to define the strength of a given linear
% regression relationship
```