Video: Scatter Plots and Linear Fits in MATLAB (5:40 min)

Show relationship on scatterplot (0:00):

Now let me look at the relationship graphically. I'm going to plot child vs parent in an x-y plot. Parent should go on the x-axis since its the independent variable. I save and execute and I see a rather jumbled plot. By default, matlab connects successive points rather than showing them as markers. I have to override that by giving the marker with no line indicator. Now I save and execute and I see a scatter plot. The markers show, but not the line. Let me finish labeling the plot. The x-axis is the parent beak size and the y-axis is the child's. Let me also put a title on this. I execute and I have a fully labeled graph.

Display the Correlation (1:19):

Let me display the correlation I calculated in the previous step as part of the title. I'm going to start by defining a new variable called tString. I'll put my original title in and i'll add the correlation. This isn't a fprintf fo I have to convert the correlation to a string in order to use it. The num2str function does that for me. Now I'm going to use tString as my title. I execute and now I see that the correlation is now a part of the title. Since I already have tString, I might as well use it to label the figure as well. Name puts the title on the title bar.

Plot Tools (2:12):

Let me use the plot tools to show the best linear before I calculate it myself. On the figure window, I use the tools window and select basic fitting. A menu will come up showing me my options. We are going to model the relationship between the parent and child as a straight line, so we check the linear fit. You see there are a lot of options. We are not going to worry about centering and scaling the data for this, but we will show the equation for the graph. Wd close the window and we see the equation. On the file menu on the figure window, I'll select save as and I'll save the figure as beaks.fig. Notice that the slope of the line is about .77 and the intercept is about 1.8. Now I'm going to do my own linear fit and compare the results.

Own Linear Fit (3:08):

The MATLAB polyfit function allows me to fit a polynomial of any degree to fit the data. I'm going to do a linear fit which is a degree 1. I type polyfit, the name of the independent variable which is parent, the name of the dependent variable which is child, and the degree. I save and execute, pPoly has the coefficients of the fitted polynomial. When I round the two digits they are 0.77 and 1.8, this agrees with the results from plot tools. The linear fit is a model that allows me to do predictions. I simply plug in the parent value into the polynomial equation and I get an estimate or prediction for the child. The matlab polyval function allows me to plug in all the parent values at once. The first argument of polyval are the coefficients i computed for my model, the second argument is the independent variable, in this case the vector parent.

Prediction Error (4:20)

I can evaluate the prediction error by taking the real values, the values in child, and subtracting them from the predicted values, the values in pPred. I define the variable pError to be this prediction error. When I evaluate I see that pError is a 22x1 vector. I'd like to get a single

number for my prediction error. To do this, I'm going to calculate the mean squared error. This is done by taking the average after squaring pError. The .* is for element by element multiplication. I evaluate and see that my error is about 0.26. This mean squared error is in units of millimeters squared, so I'm going to square root it to get it in units of millimeters, this is called the root mean squared error. I use the square root function to calculate it. When I evaluate it I see my root mean squared error is about 0.5 millimeters.