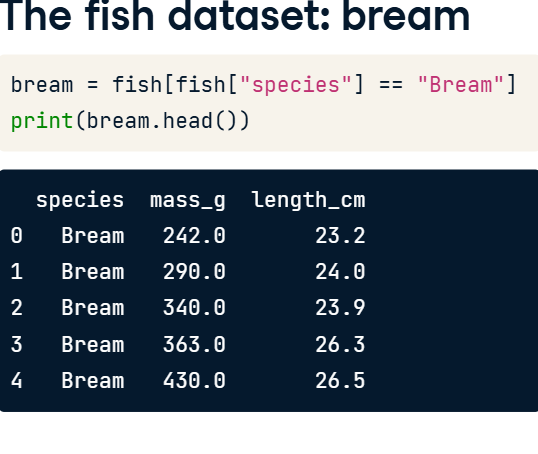
**Making predictions**

The big benefit of running models rather than simply calculating descriptive statistics is that models let you make predictions.**The fish dataset: bream**

Here's the fish dataset again. This time, we'll look only at the bream data. There's a new explanatory variable too: the length of each fish, which we'll use to predict the mass of the fish.

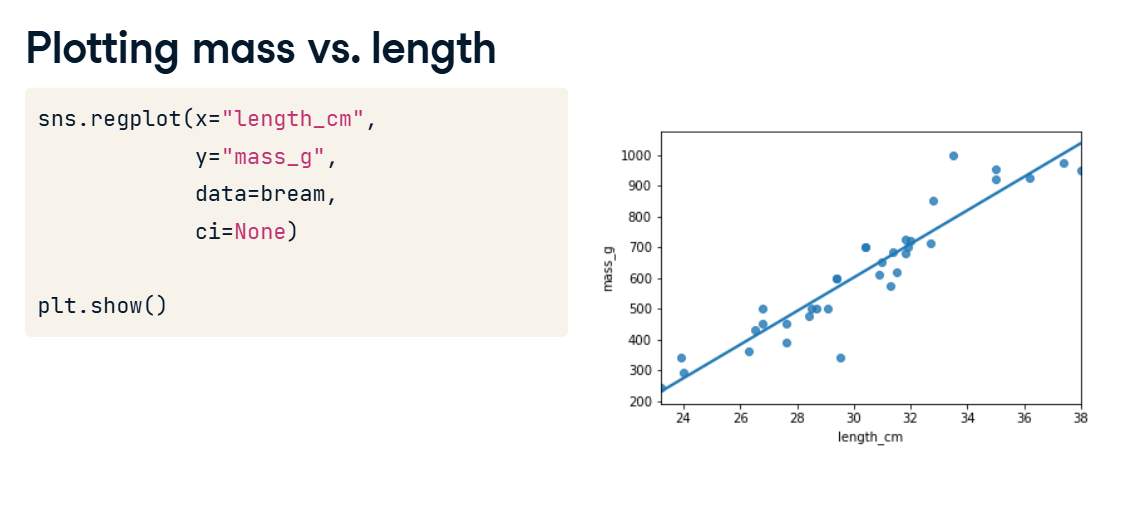


**3. Plotting mass vs. length**

Here's a scatter plot of mass versus length for the bream data, with a linear trend line.

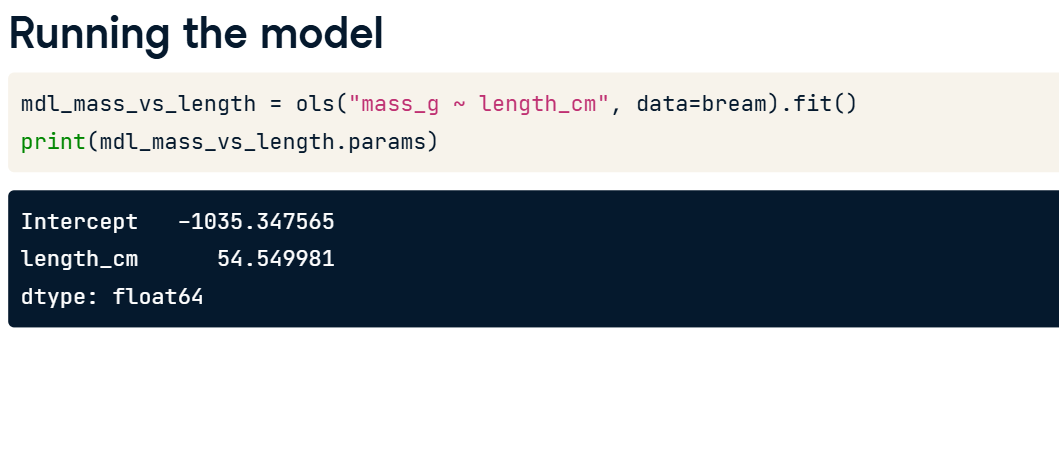
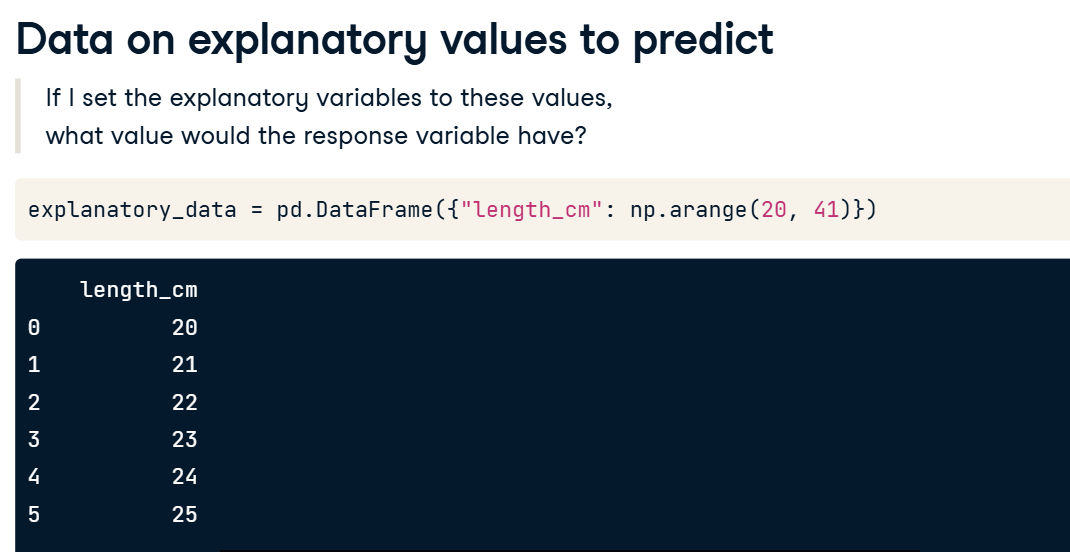
**Running the model**

Before we can make predictions, we need a fitted model. As before, we call ols with a formula and the dataset, after which we add dot fit. The response, mass in grams, goes on the left-hand side of the formula, and the explanatory variable, length in centimeters, goes on the right. We need to assign the result to a variable to reuse later on. To view the coefficients of the model, we use the params attribute in a print call.



**5. Data on explanatory values to predict**

The principle behind predicting is to ask questions of the form "if I set the explanatory variables to these values, what value would the response variable have?". That means that the next step is to choose some values for the explanatory variables. To create new explanatory data, we need to store our explanatory variables of choice in a pandas DataFrame. You can use a dictionary to specify the columns. For this model, the only explanatory variable is the length of the fish. You can specify an interval of values using the np dot arange function, taking the start and end of the interval as arguments. Notice that the end of the interval does not include this value. Here, I specified a range of twenty to forty centimeters.



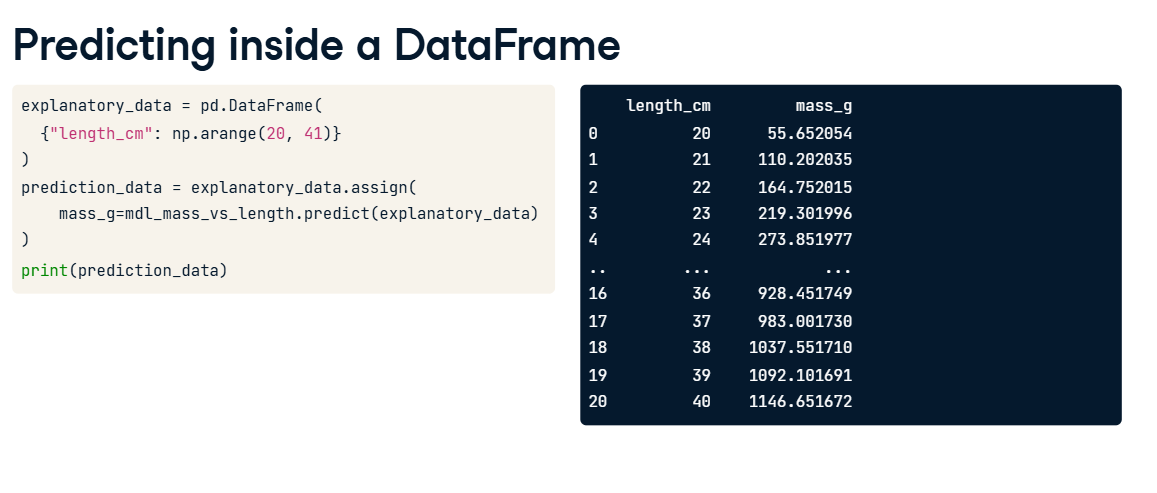
**6. Call predict()**

The next step is to call predict on the model, passing the DataFrame of explanatory variables as the argument. The predict function returns a Series of predictions, one for each row of the explanatory data.



**7. Predicting inside a DataFrame**

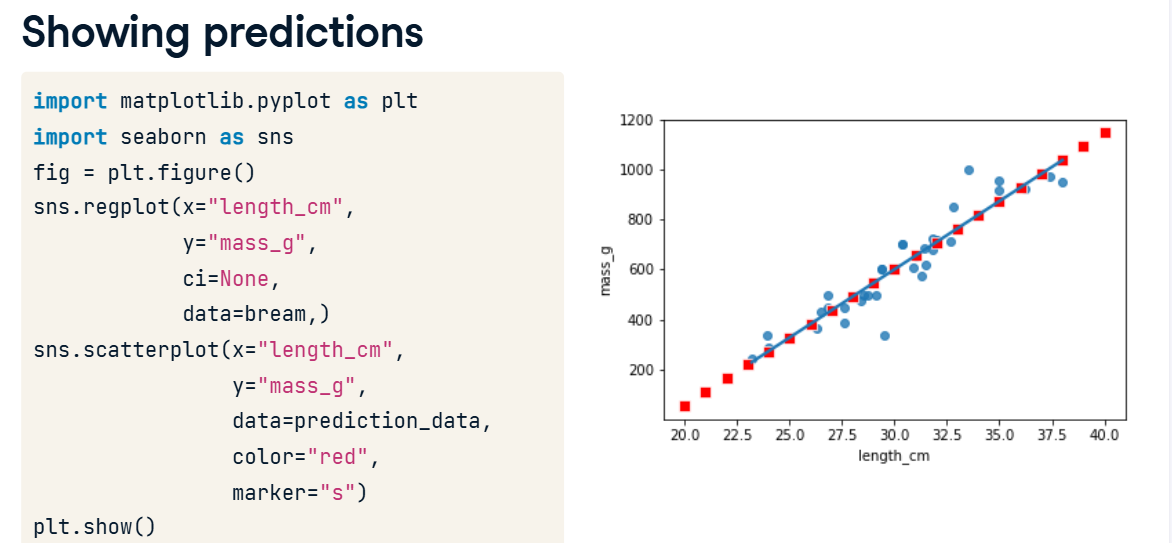
Having a single column of predictions isn't that helpful to work with. It's easier to work with if the predictions are in a DataFrame alongside the explanatory variables. To do this, you can use the pandas assign method. It returns a new object with all original columns in addition to new ones. You start with the existing column, explanatory\_data. Then, you use dot assign to add a new column, named after the response variable, mass\_g. You calculate it with the same predict code from the previous slide. The resulting DataFrame contains both the explanatory variable and the predicted response. Now we can answer questions like "how heavy would we expect a bream with length twenty three centimeters to be?", even though the original dataset didn't include a bream of that exact length. Looking at the prediction data, you can see that the predicted mass is two hundred and nineteen grams.



**8. Showing predictions**

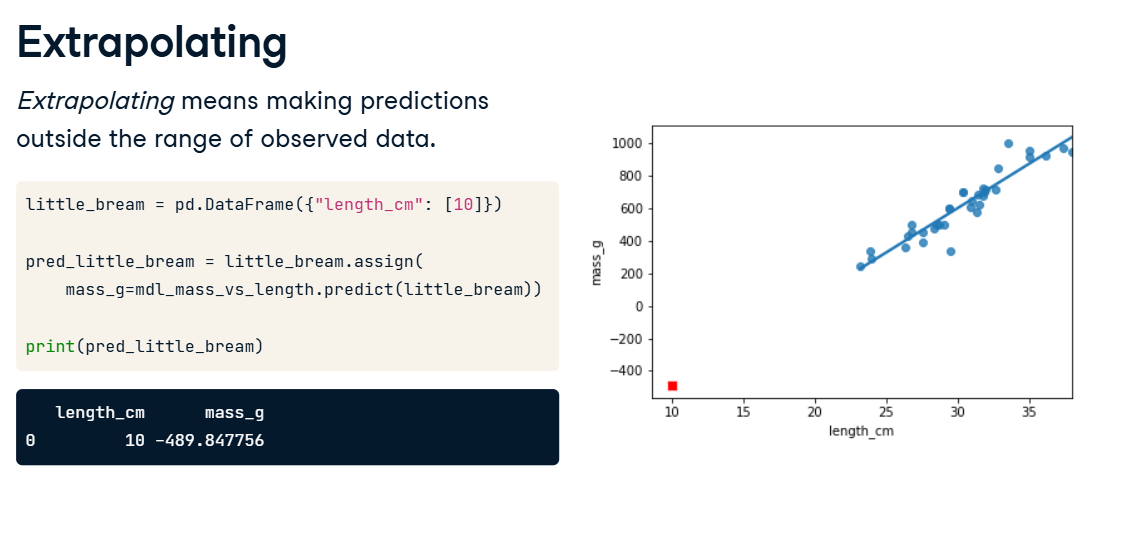
03:09 - 03:39

Let's include the predictions we just made on the scatter plot. To plot multiple layers, we set a matplotlib figure object called fig before calling regplot and scatterplot. As a result, the plt dot show call will then plot both graphs on the same figure. I've marked the prediction points in red squares to distinguish them from the actual data points. Notice that the predictions lie exactly on the trend line.



**9. Extrapolating**

All the fish were between twenty three and thirty eight centimeters, but the linear model allows us to make predictions outside that range. This is called extrapolating. Let's see what prediction we get for a ten centimeter bream. To achieve this, you first create a DataFrame with a single observation of 10 cm. You then predict the corresponding mass as before. Wow. The predicted mass is almost minus five hundred grams! This is obviously not physically possible, so the model performs poorly here. Extrapolation is sometimes appropriate, but can lead to misleading or ridiculous results. You need to understand the context of your data in order to determine whether it is sensible to extrapolate.

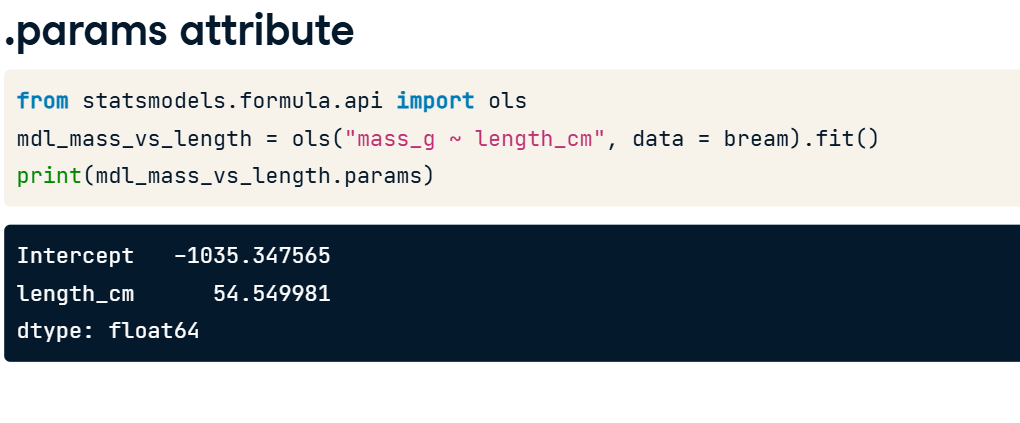


**Working with model objects**

The model objects created by ols contain a lot of information. In this video, you'll see how to extract it.

**2. .params attribute**

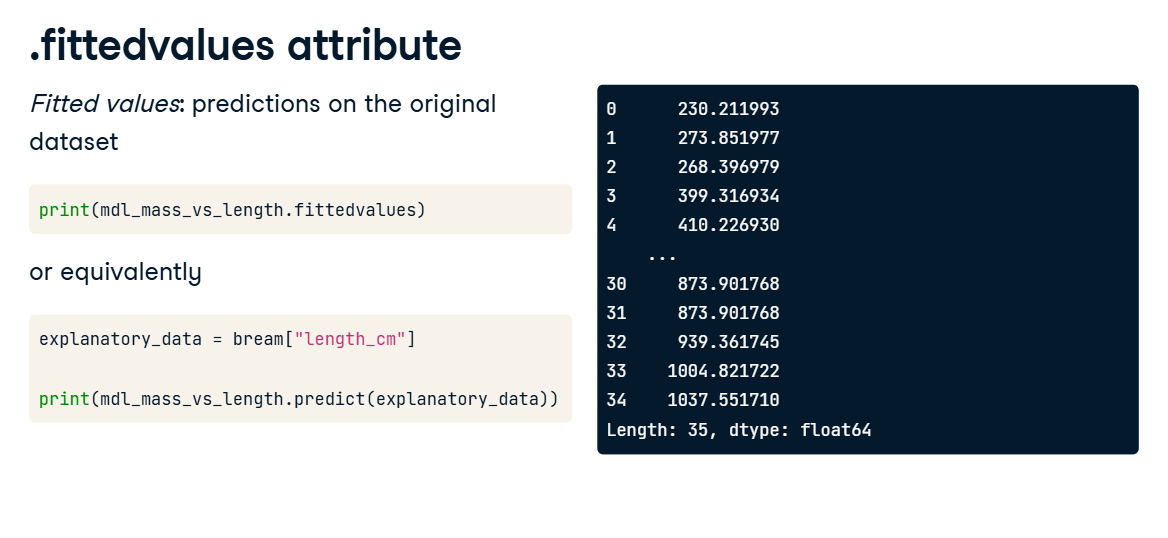
You already learned how to extract the coefficients or parameters from your fitted model. You add the dot params attribute, which will return a pandas Series including your intercept and slope.



**3. .fittedvalues attribute**

00:21 - 00:48

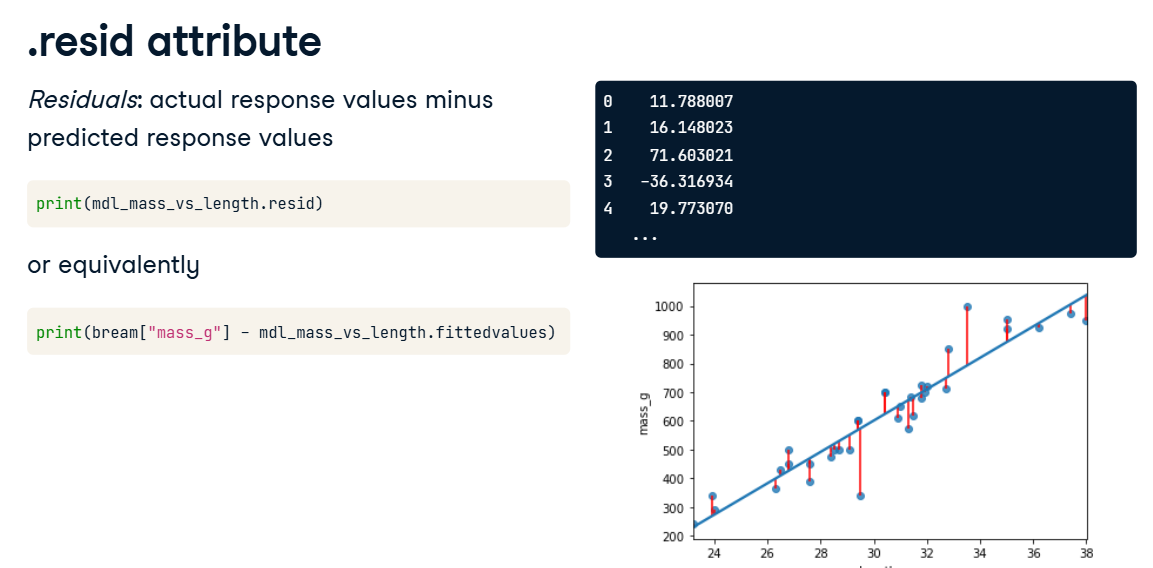
"Fitted values" is jargon for predictions on the original dataset used to create the model. Access them with the fittedvalues attribute. The result is a pandas Series of length thirty five, which is the number of rows in the bream dataset. The fittedvalues attribute is essentially a shortcut for taking the explanatory variable columns from the dataset, then feeding them to the predict function.



**4. .resid attribute**

00:48 - 01:28

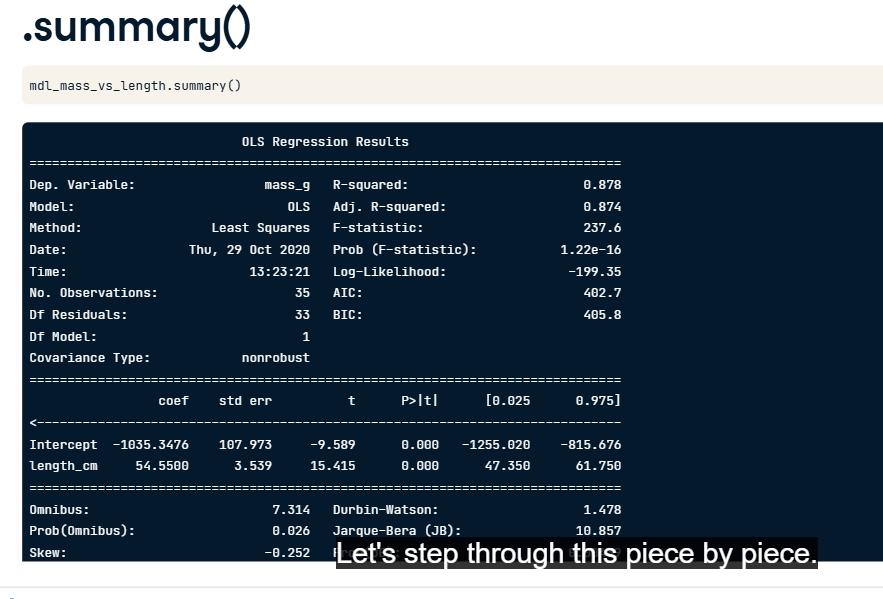
"Residuals" are a measure of inaccuracy in the model fit, and are accessed with the resid attribute. Like fitted values, there is one residual for each row of the dataset. Each residual is the actual response value minus the predicted response value. In this case, the residuals are the masses of breams, minus the fitted values. I illustrated the residuals as red lines on the regression plot. Each vertical line represents a single residual. You'll see more on how to use the fitted values and residuals to assess the quality of your model in Chapter 3.



**5. .summary()**

01:28 - 01:37

The summary method shows a more extended printout of the details of the model. Let's step through this piece by piece.



**6. .summary() part 1**

01:37 - 01:50

First, you see the dependent variable(s) that were used in the model, in addition to the type of regression. You also see some metrics on the performance of the model. These will be discussed in the next chapter.

**7. summary() part 2**

In the second part of the summary, you see details of the coefficients. The numbers in the first column are the ones contained in the params attribute. The numbers in the fourth column are the p-values, which refer to statistical significance. You can learn about them in DataCamp's courses on inference. The last part of the summary are diagnostic statistics that are outside the scope of this course.