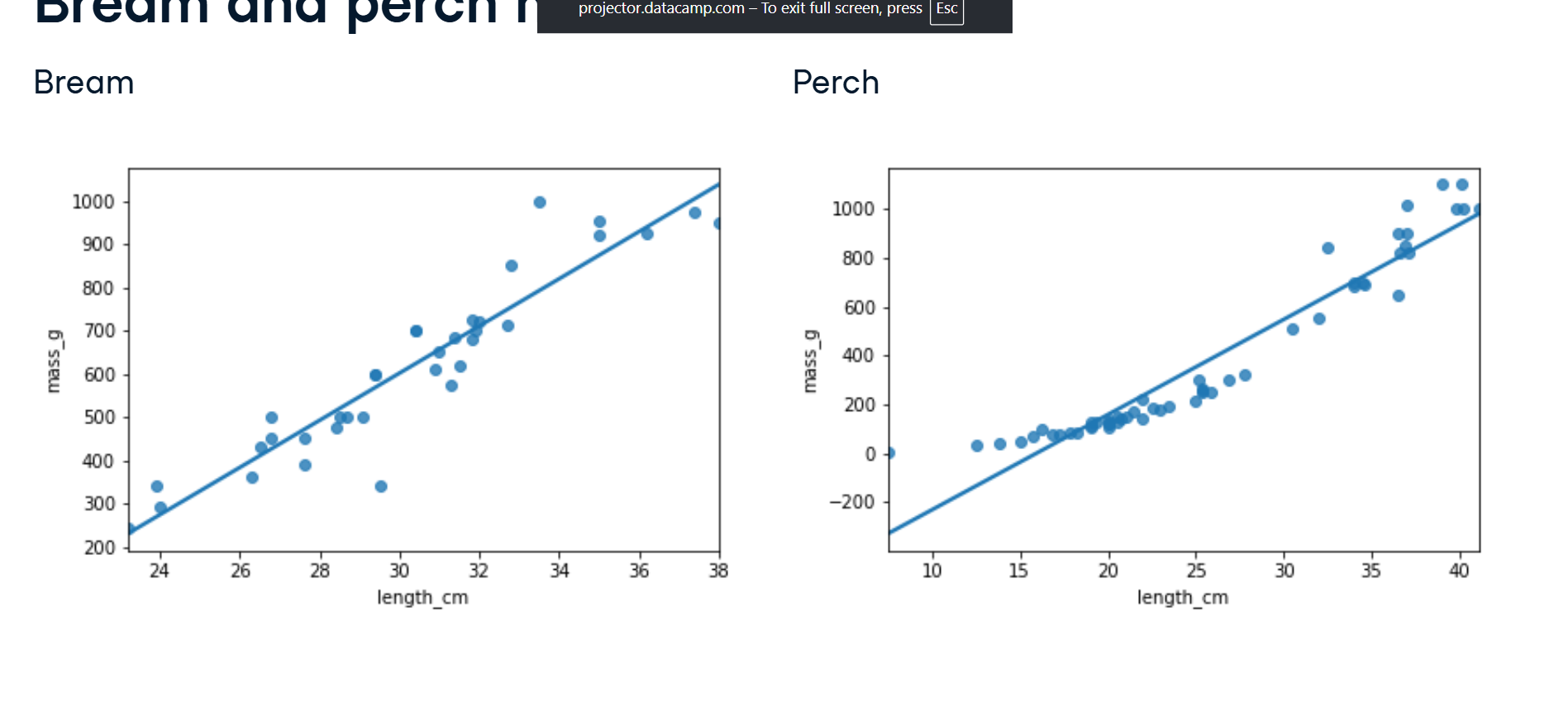
**Quantifying model fit**

It's usually essential to know whether or not predictions from your model are nonsense. In this chapter, we'll look at ways of quantifying how good your model is.

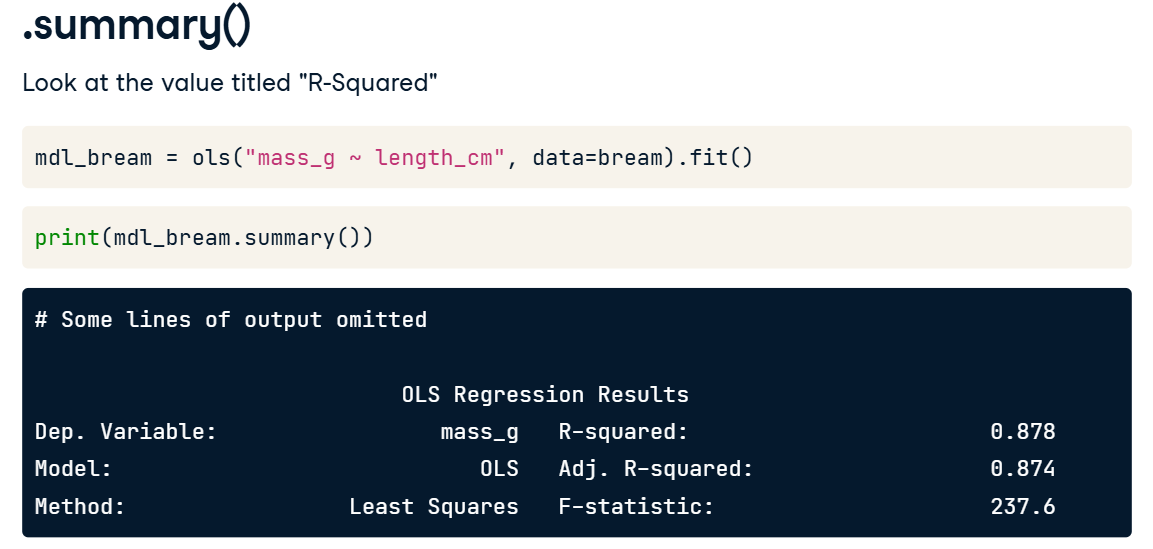
**2. Bream and perch models**

Previously, you ran models on mass versus length for bream and perch. By merely looking at these scatter plots, you can get a sense that there is a linear relationship between mass and length for bream but not for perch. It would be useful to quantify how strong that linear relationship is.



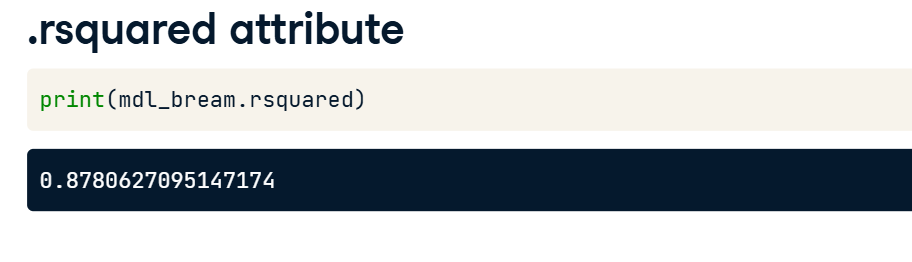
**3. Coefficient of determination**

The first metric we'll discuss is the coefficient of determination. This is sometimes called "r-squared". For boring historical reasons, it's written with a lower case r for simple linear regression and an upper case R when you have more than one explanatory variable. It is defined as the proportion of the variance in the response variable that is predictable from the explanatory variable. We'll get to a human-readable explanation shortly. A score of one means you have a perfect fit, and a score of zero means your model is no better than randomness. What constitutes a good score depends on your dataset. A score of zero-point five on a psychological experiment may be exceptionally high because humans are inherently hard to predict, but in other cases, a score of zero-point nine may be considered a poor fit.



**4. .summary()**

The dot summary method shows several performance metrics in its output. The coefficient of determination is written in the first line and titled "R-squared". Its value is about zero-point-eight-eight.

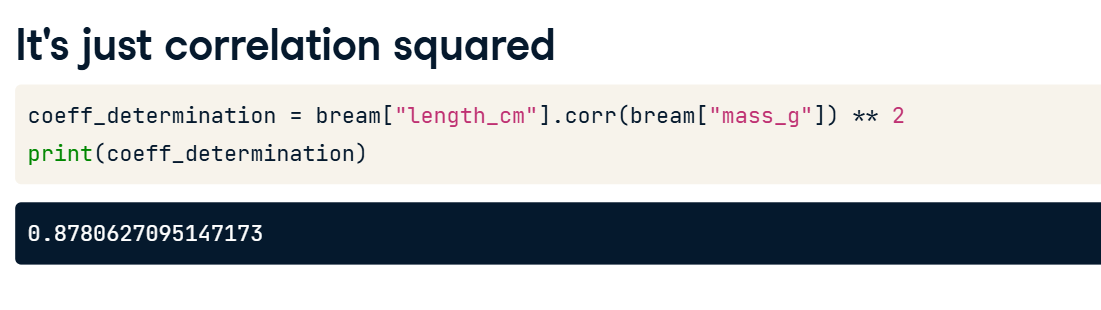


**5. .rsquared attribute**

Since the output of dot summary isn't easy to work with, a better way to extract the metric is to use the rsquared attribute, which contains the r-squared value as a float.

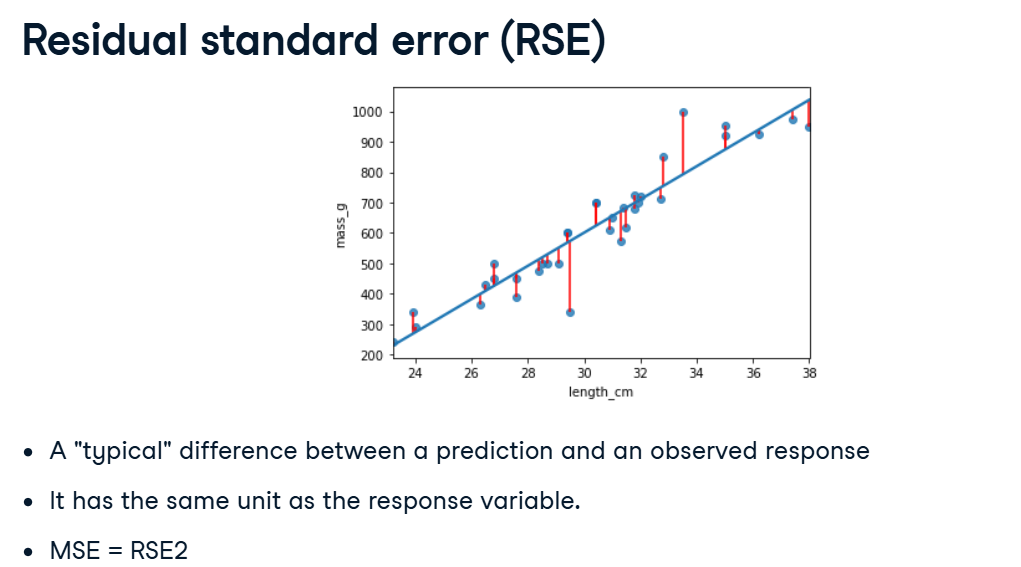
**6. It's just correlation squared**

For simple linear regression, the interpretation of the coefficient of determination is straightforward. It is simply the correlation between the explanatory and response variables, squared.



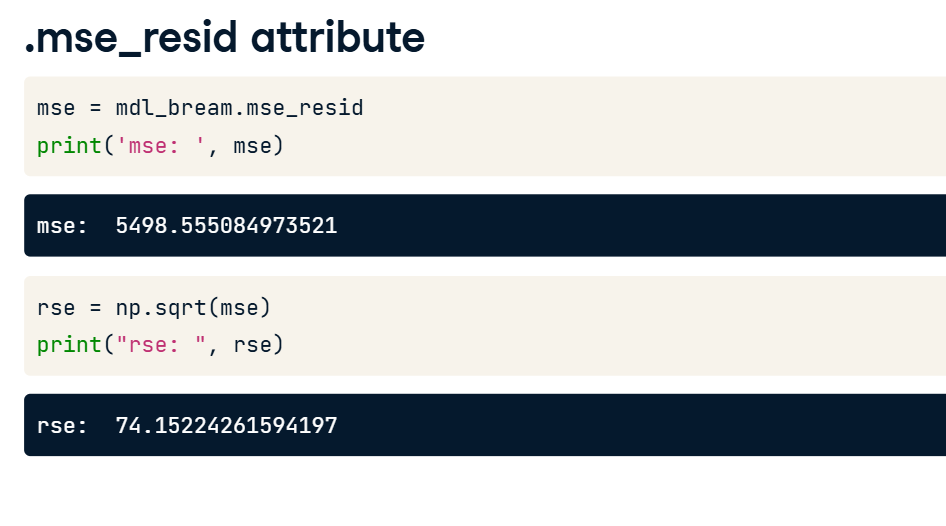
**7. Residual standard error (RSE)**

The second metric we'll look at is the residual standard error, or RSE. Recall that each residual is the difference between a predicted value and an observed value. The RSE is, very roughly speaking, a measure of the typical size of the residuals. That is, how much the predictions are typically wrong. It has the same unit as the response variable. In the fish models, the response unit is grams. A related, but less commonly used metric is the mean squared error, or MSE. As the name suggests, MSE is the squared residual standard error.



**8. .mse\_resid attribute**

The summary method unfortunately doesn't contain the RSE. However, it can indirectly be retrieved from the mse\_resid attribute, which contains the mean squared error of the residuals. We can calculate the RSE by taking the square root of MSE. As such, the RSE has the same unit as the response variable. The RSE for the bream model is about seventy-four.



**9. Calculating RSE: residuals squared**

To calculate the RSE yourself, it's slightly more complicated. First, you take the square of each residual.

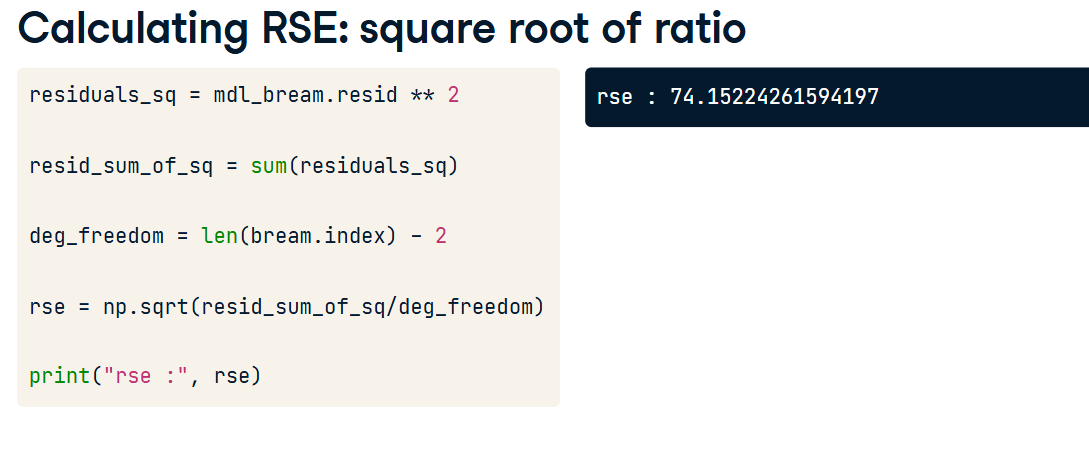
Then you take the sum of these residuals squared.

You then calculate the degrees of freedom of the residuals. This is the number of observations minus the number of model coefficients.



**12. Calculating RSE: square root of ratio**

Finally, you take the square root of the ratio of those two numbers. Reassuringly, the value is still seventy-four.

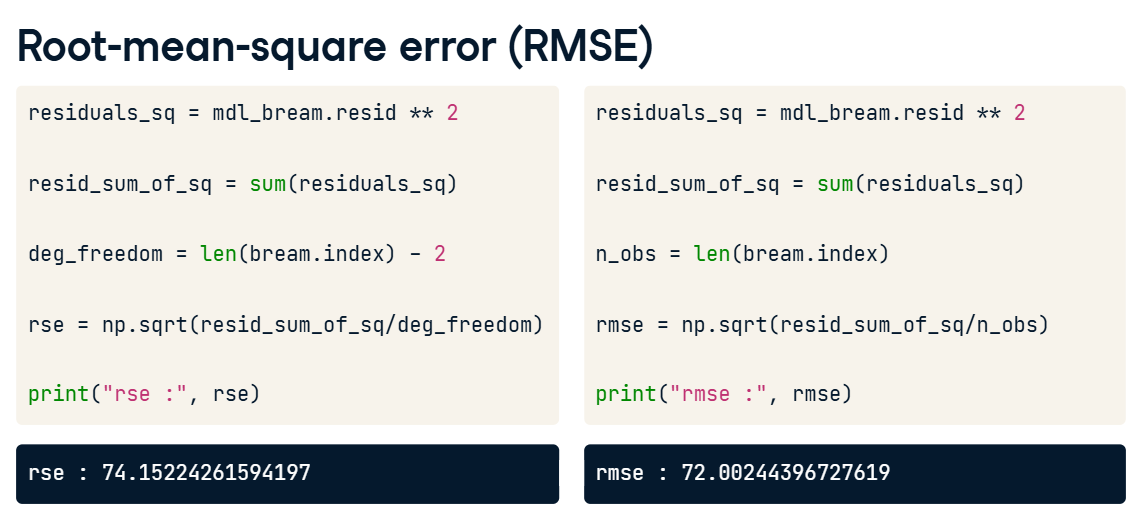


**13. Interpreting RSE**

An RSE of seventy-four means that the difference between predicted bream masses and observed bream masses is typically about seventy-four grams.

**14. Root-mean-square error (RMSE)**

Another related metric is the root-mean-square error. This is calculated in the same way, except you don't subtract the number of coefficients in the second to last step. It performs the same task as residual standard error, namely quantifying how inaccurate the model predictions are, but is worse for comparisons between models. You need to be aware that RMSE exists, but typically you should use RSE instead.



**Visualizing model fit**

Several plots can quantify the performance of a model. We'll look at these plots and their interpretation first, then the code to draw them.

**2. Residual properties of a good fit**

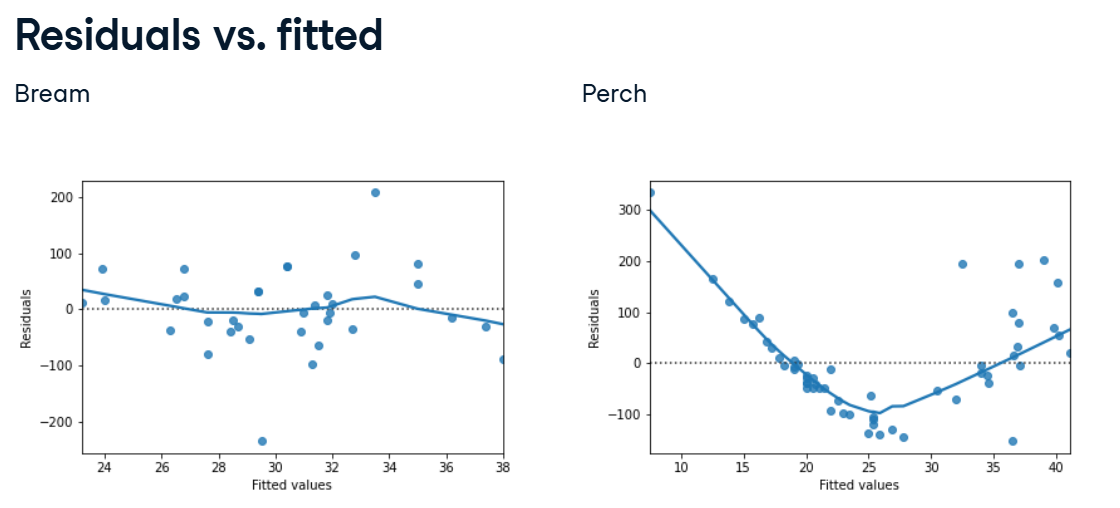
If a linear regression model is a good fit, then the residuals are approximately normally distributed, with mean zero.

**3. Bream and perch again**

Earlier, we ran models on the bream and perch datasets. From looking at the scatter plots with linear trend lines, it appeared that the bream model was a good fit, but the perch model wasn't because the observed masses increased faster than linearly with the lengths.

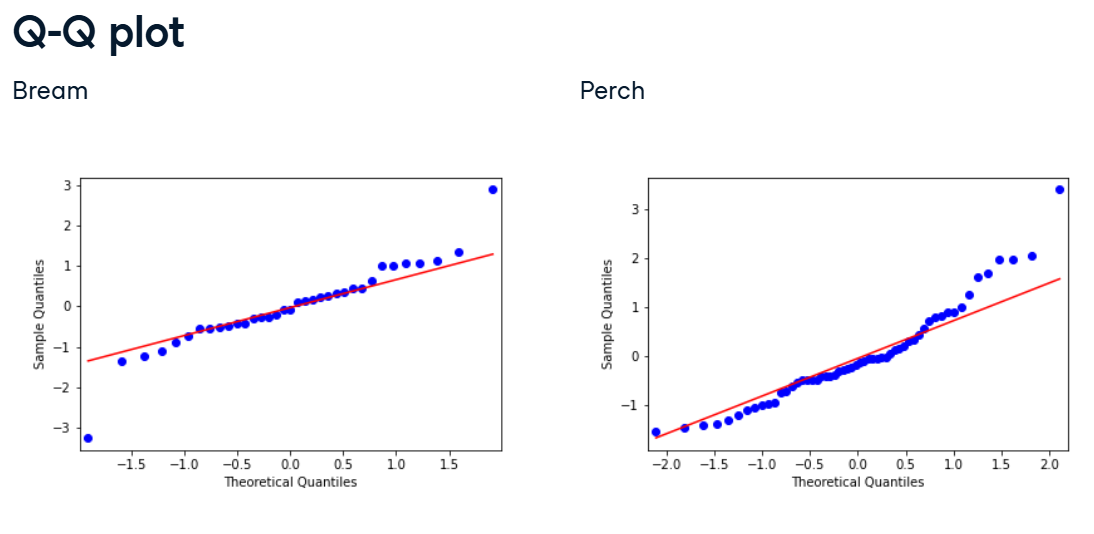
**4. Residuals vs. fitted**

The first diagnostic plot is of residuals versus fitted values. The blue line is a LOWESS trend line, which is a smooth curve following the data. These aren't good for making predictions but are useful for visualizing trends. If residuals met the assumption that they are normally distributed with mean zero, then the trend line should closely follow the y equals zero line on the plot. For the bream dataset, this is true. By contrast, the perch model doesn't meet the assumption. The residuals are above zero when the fitted value is small or big and below zero in the middle.



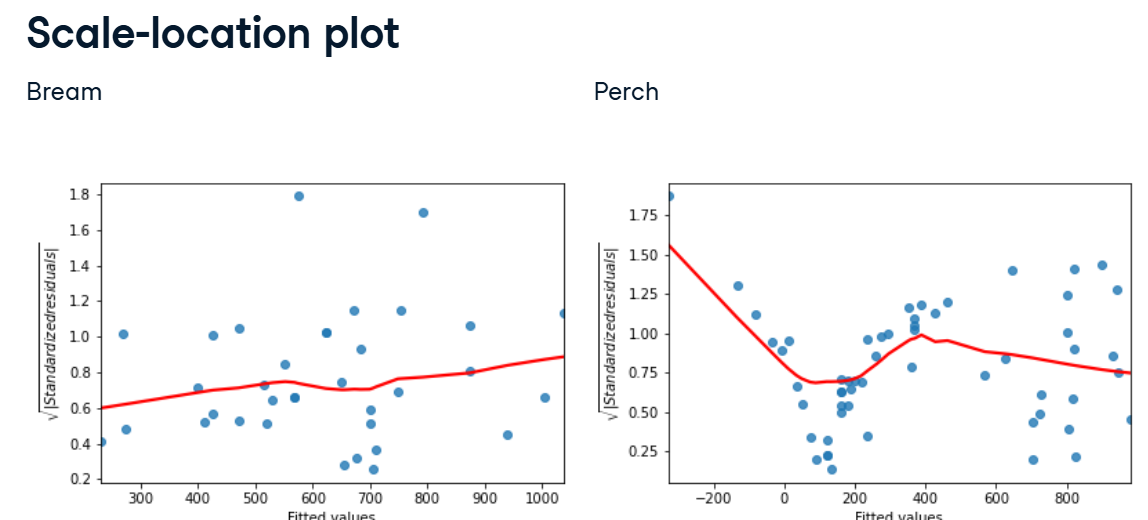
**5. Q-Q plot**

The second diagnostic plot is called a Q-Q plot. It shows whether or not the residuals follow a normal distribution. On the x-axis, the points are quantiles from the normal distribution. On the y-axis, you get the sample quantiles, which are the quantiles derived from your dataset. It sounds technical, but interpreting this plot is straightforward. If the points track along the straight line, they are normally distributed. If not, they aren't. Here, most of the bream points follow the line closely. Two points at each extreme don't follow the line. These correspond to the rows of the bream dataset with the highest residuals. The perch dataset doesn't track the line as closely. In particular, you can see on the right-hand side of the plot that the residuals are larger than expected. That means the model is a particularly poor fit for the longer lengths of perch.



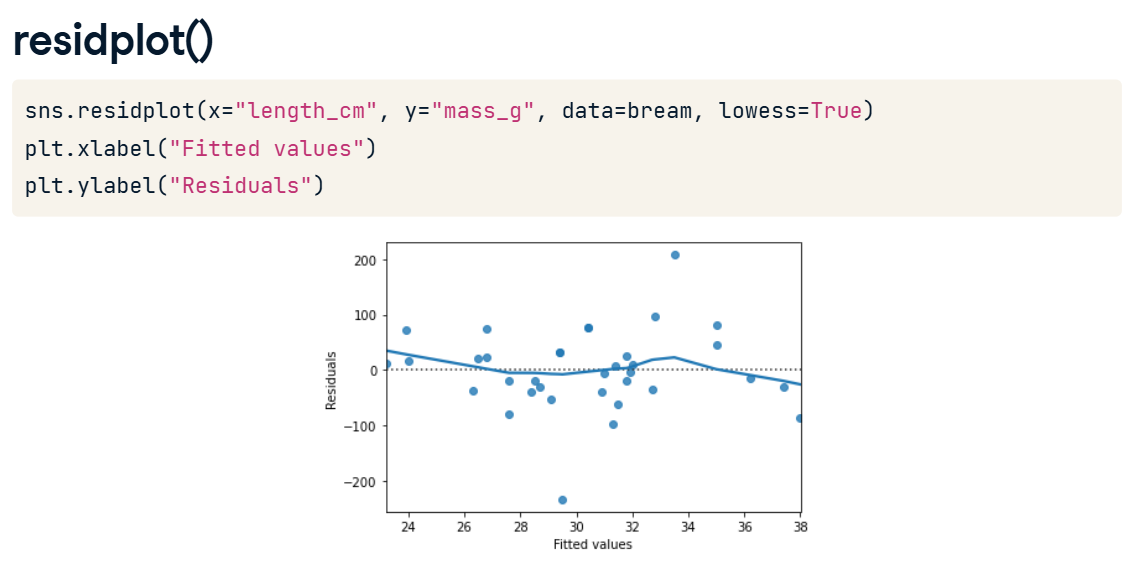
**6. Scale-location plot**

The third plot shows the square root of the standardized residuals versus the fitted values. It's often called a scale-location plot, because that's easier to say. Where the first plot showed whether or not the residuals go positive or negative as the fitted values change, this plot shows whether the size of the residuals gets bigger or smaller. The residuals for the bream dataset get a little bigger as the fitted values increase, but it's not a huge change. Again, the plot of the perch model has a trend line that goes up and down all over the place, indicating a poor fit.



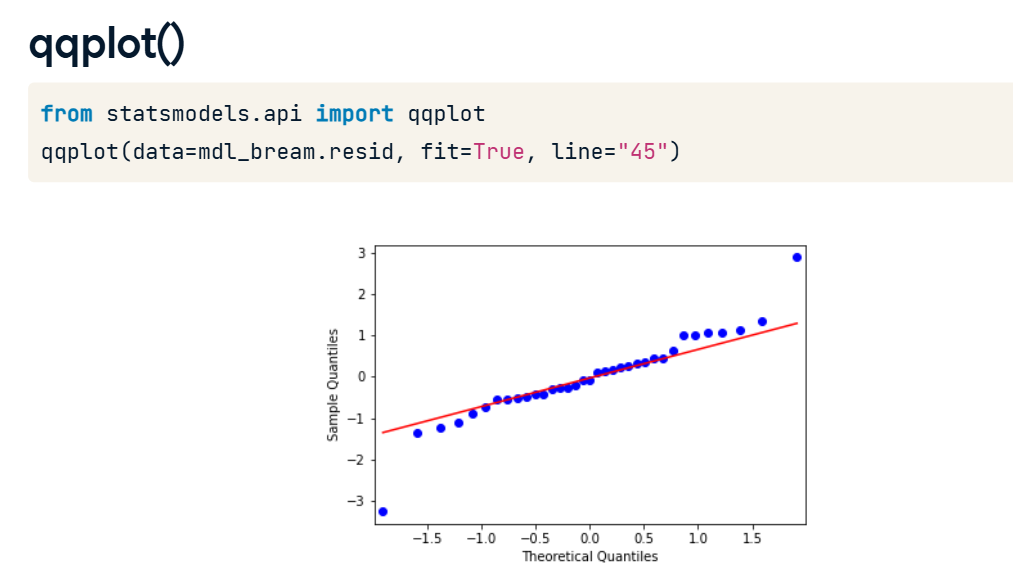
**7. residplot()**

To create the residuals vs. fitted plot, you can use the residplot function from seaborn. It takes the usual x, y, and data arguments, in addition to the lowess argument. This will add a smooth curve following the data, visualizing the trend of your residuals. You'll also need to specify the x and y labels manually.



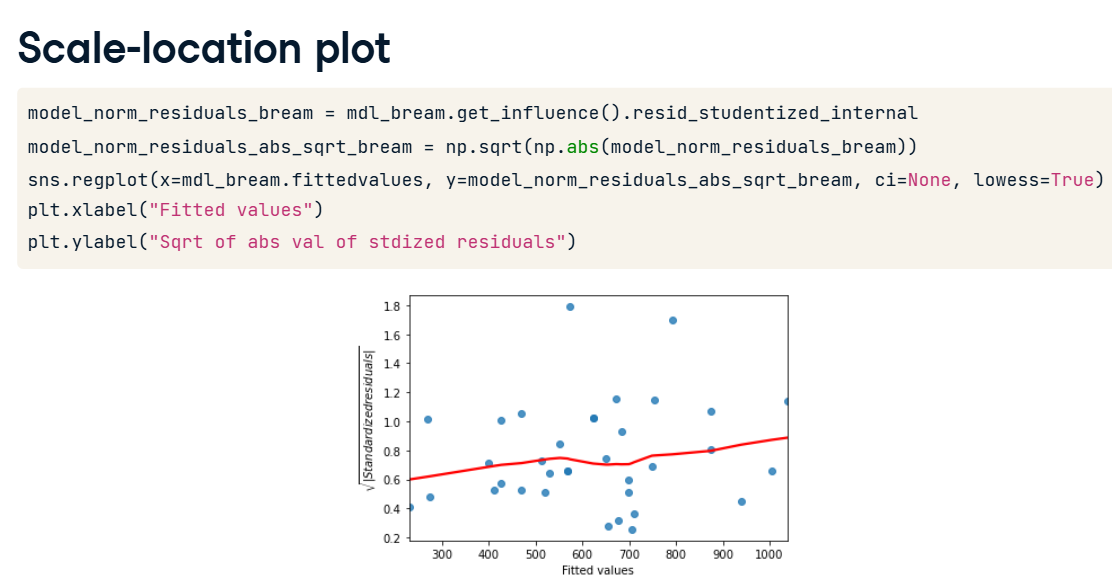
**8. qqplot()**

To draw a Q-Q plot, you can use the qqplot function from the statsmodels package. You set the residuals of the model as your data argument and the fit argument to True. This will compare the data quantiles to a normal distribution. The last argument is optional, but when set to "45", set as a string, it will draw a 45-degree line on your plot, making it easier to interpret the pattern.



**9. Scale-location plot**

The last plot, scale-location, requires a bit more preprocessing. You first need to extract the normalized residuals from the model, which you can get by using the get\_influence method, then accessing the resid\_studentized\_internal attribute. Don't worry about this too much now, we'll come back to that in the following lesson. You then take the absolute values and take the square root of these normalized residuals to standardize them. Next, you can call sns dot regplot, passing in mdl\_bream dot fittedvalues for x, and the standardized residuals for y. Again, you can also include a lowess argument to make interpretation easier. Lastly, you specify the axes manually.



**Outliers, leverage, and influence**

Sometimes, datasets contain unusual values. We'll look at how to spot them and the consequences they have for your regression models.

**2. Roach dataset**

00:09 - 00:15

Let's look at another species in the fish dataset, this time filtering for the Common roach.

**3. Which points are outliers?**

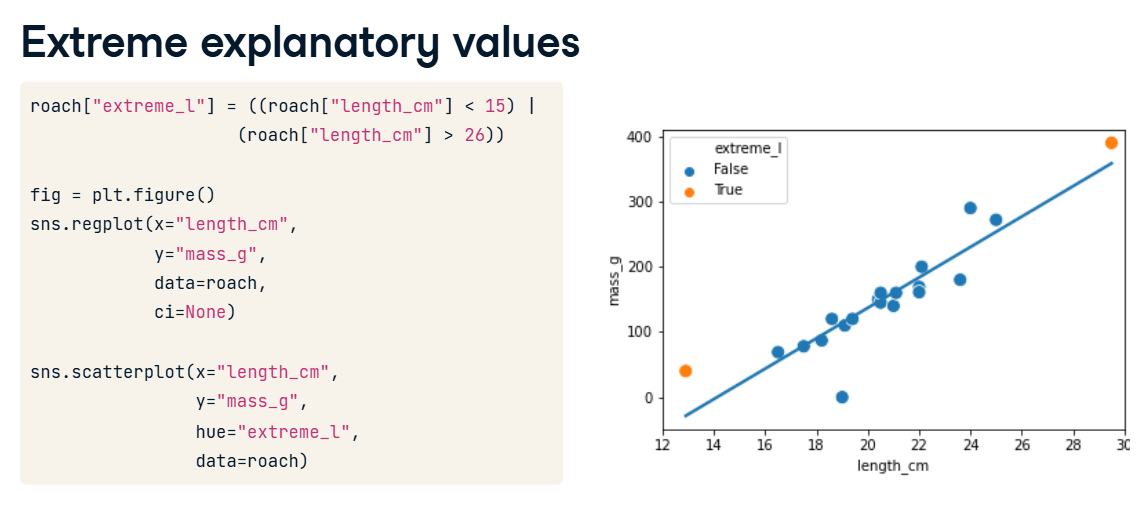
00:15 - 00:27

Here's the standard regression plot of mass versus length. The technical term for an unusual data point is an outlier. So which of these points constitutes an outlier?

**4. Extreme explanatory values**

00:27 - 00:43

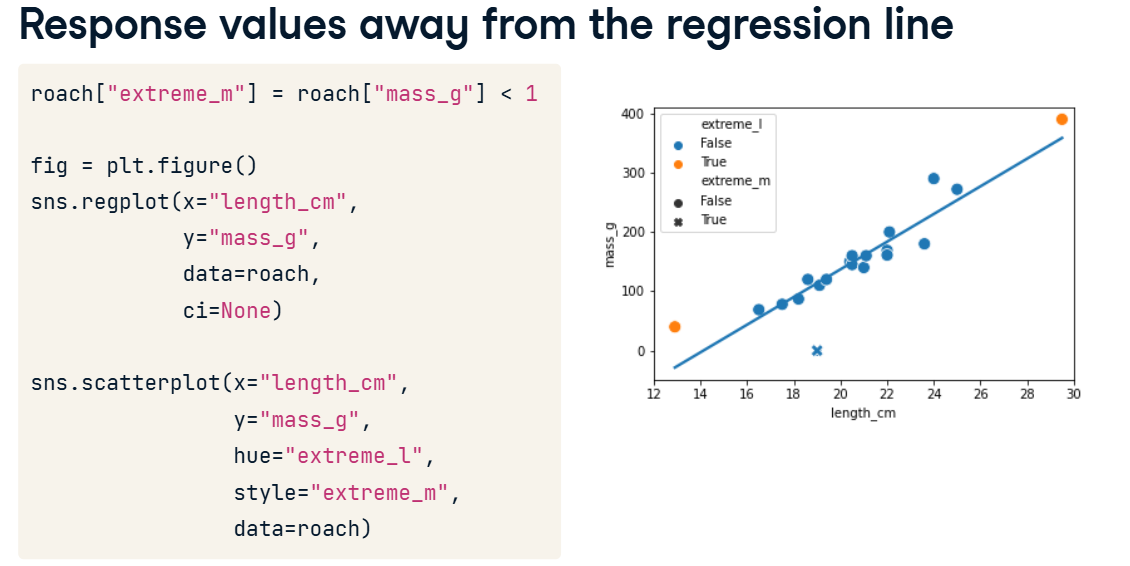
The first kind of outlier is when you have explanatory variables that are extreme. In the simple linear regression case, it's easy to find and visualize them. There is one extreme short roach and one extreme long roach that I've colored orange here.



**5. Response values away from the regression line**

00:43 - 00:56

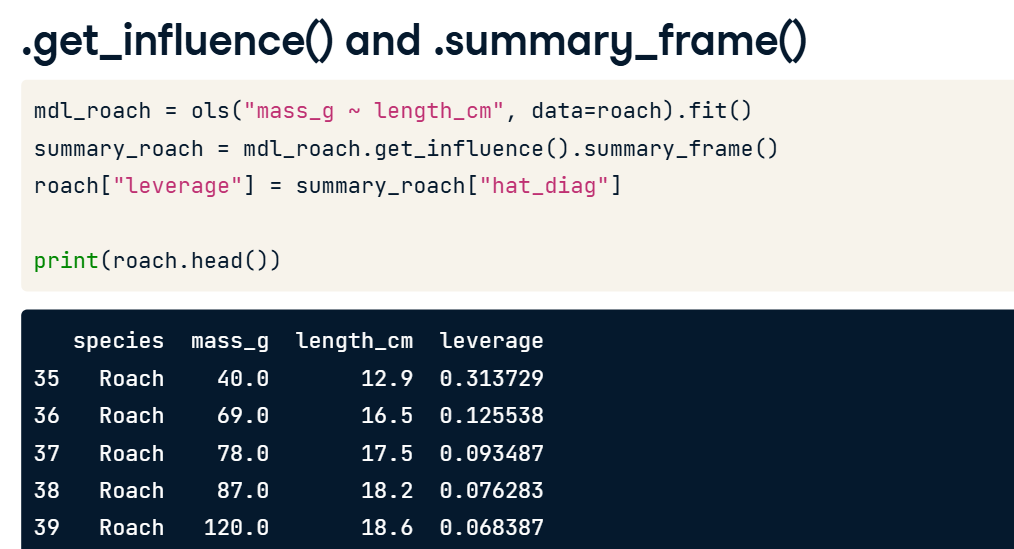
The other property of outliers is when the point lies a long way from the model predictions. Here, there's a roach with mass zero, which seems biologically unlikely. It's shown as a cross.



**6. Leverage and influence**

00:56 - 01:47

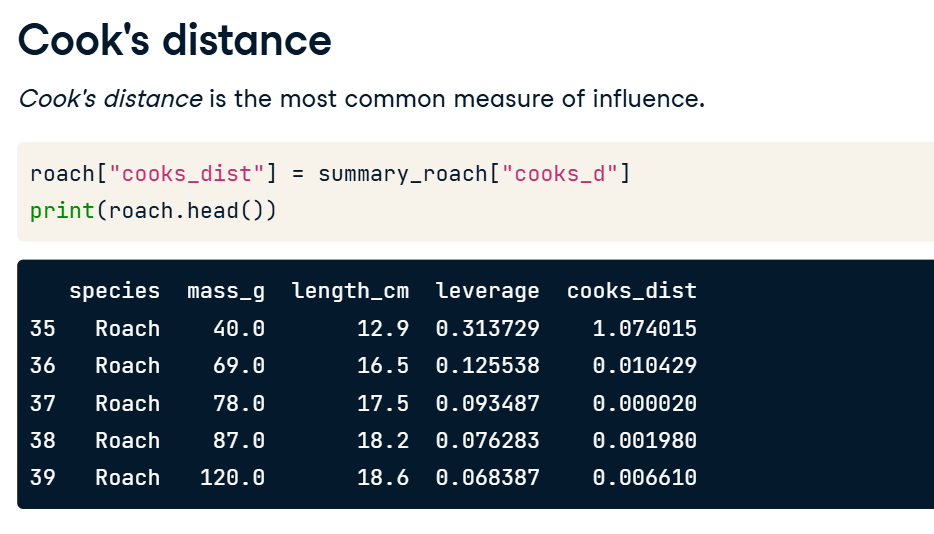
Leverage quantifies how extreme your explanatory variable values are. That is, it measures the first type of outlier we discussed. With one explanatory variable, you can find the values by filtering, but with many explanatory variables, the mathematics is more complicated. A related concept to leverage is influence. This is a type of "leave one out" metric. That is, it measures how much the model would change if you reran it without that data point. I like to think of it as the torque of the point. The amount of turning force, or torque, when using a wrench is equal to the linear force times the length of the wrench. In a similar way, the influence of each observation is based on the size of the residuals and the leverage.



**7. .get\_influence() and .summary\_frame()**

01:47 - 02:26

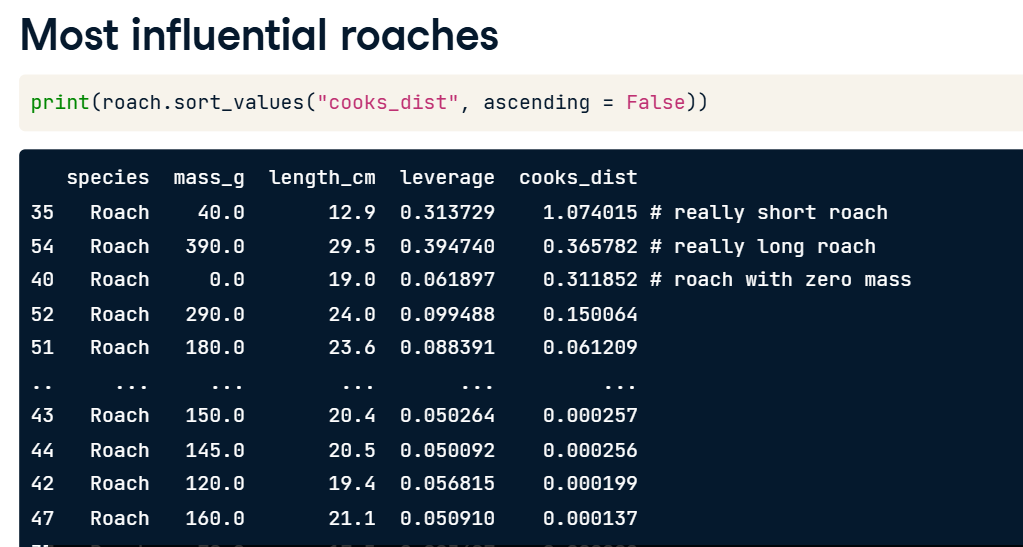
Leverage and influence, along with other metrics, are retrieved from the summary frame. You get them by calling the get\_influence() method on the fitted model, then calling the summary\_frame() method. For historical reasons, leverage is described in the so-called hat matrix. Therefore, the values of leverage are stored in the hat\_diag column of the summary frame. Like the fitted values and residuals methods, it returns an array with as many values as there are observations. In this case, each of these leverage values indicates how extreme your roach lengths are.



**8. Cook's distance**

02:26 - 02:41

Recall that influence is based on the size of the residuals and the leverage. It isn't a straightforward multiplication; instead, we use a metric called Cook's distance. It is stored in the summary frame as 'cooks\_d'.



**9. Most influential roaches**

02:41 - 02:56

We can find the most influential roaches by arranging the rows by descending Cook's distance values. Here, you can see the two highly leveraged points and the fish with zero mass that gave it a large residual.

**10. Removing the most influential roach**

To see how influence works, let's remove the most influential roach. This is the one with the shortest length, at twelve-point-nine centimeters. We draw the usual regression plot but add another regression line using the dataset without that short fish. The slope of the line has completely changed just by having one less data point.

