**Transcript**

**1. Annotating time-series data**

00:00 - 00:19

One important way to enhance a visualization is to add annotations. Annotations are usually small pieces of text that refer to a particular part of the visualization, focusing our attention on some feature of the data and explaining this feature.

**2. Time-series data**

00:19 - 00:50

For example, consider the data that we saw in previous videos in this chapter. This data shows the levels of measured carbon dioxide in the atmosphere over a period of more than 50 years in blue and the relative temperature over the same period of time in red. That's a lot of data, and, when presenting it, you might want to focus attention on a particular aspect of this data.

**3. Annotation**

00:50 - 02:19

One way to draw attention to part of a plot is by annotating it. This means drawing an arrow that points to part of the plot and being able to include text to explain it. For example, let's say that we noticed that the first date in which the relative temperature exceeded 1 degree Celsius was October 6th, 2015. We'd like to point this out in the plot. Here again is the code that generates the plot, using the function that we implemented previously. Next, we call a method of the Axes object called annotate. At the very least, this function takes the annotation text as input, in this case, the string ">1 degree", and the xy coordinate that we would like to annotate. Here, the value to annotate has the x position of the TimeStamp of that date. We use the pandas time-stamp object to define that. The y position of the data is 1, which is the 1 degree Celsius value at that date. But this doesn't look great. The text appears on top of the axis tick labels. Maybe we can move it somewhere else?

**4. Positioning the text**

02:19 - 02:58

The annotate method takes an optional xy text argument that selects the xy position of the text. After some experimentation, we've found that an x value of October 6th, 2008 and a y value of negative 0-point-2 degrees is a good place to put the text. The problem now is that there is no way to see which data point is the one that is being annotated. Let's add an arrow that connects the text to the data.

**5. Adding arrows to annotation**

02:58 - 03:34

To connect between the annotation text and the annotated data, we can add an arrow. The key-word argument to do this is called arrowprops, which stands for arrow properties. This key-word argument takes as input a dictionary that defines the properties of the arrow that we would like to use. If we pass an empty dictionary into the key-word argument, the arrow will have the default properties, as shown here.

**6. Customizing arrow properties**

03:34 - 03:58

We can also customize the appearance of the arrow. For example, here we set the style of the arrow to be a thin line with a wide head. That's what the string with a dash and a smaller than sign means. We also set the color to gray. This is a bit more subtle.

**7. Customizing annotations**

03:58 - 04:11

There are many more options for customizing the arrow properties and other properties of the annotation, which you can read about in the Matplotlib documentation here.

**8. Practice annotating plots!**

04:11 - 04:19

But for now, start by practicing what you have learned so far.