

## **1. Introduction to Data Visualization with Matplotlib**

00:00 - 00:18

Hello and welcome to this course on data visualization with Matplotlib! A picture is worth a thousand words. Data visualizations let you derive insights from data and let you communicate about the data with others.

## **2. Data visualization**

00:18 - 01:12

For example, this visualization shows an animated history of an outbreak of Ebola in West Africa. The amount of information in this complex visualization is simply staggering! This visualization was created using Matplotlib, a Python library that is widely used to visualize data. There are many software libraries that visualize data. One of the main advantages of Matplotlib is that it gives you complete control over the properties of your plot. This allows you to customize and control the precise properties of your visualizations. At the end of this course, you will know not only how to control your visualizations, but also how to create programs that automatically create visualizations based on your data.

## **3. Introducing the pyplot interface**

01:12 - 02:19

There are many different ways to use Matplotlib. In this course, we will use the main object-oriented interface. This interface is provided through the pyplot submodule. Here, we import this submodule and name it plt. While using the name plt is not necessary for the program to work, this is a very strongly-followed convention, and we will follow it here as well. The plt-dot-subplots command, when called without any inputs, creates two different objects: a Figure object and an Axes object. The Figure object is a container that holds everything that you see on the page. Meanwhile, the Axes is the part of the page that holds the data. It is the canvas on which we will draw with our data, to visualize it. Here, you can see a Figure with empty Axes. No data has been added yet.

## **4. Adding data to axes**

02:19 - 02:50

Let's add some data to our figure. Here is some data. This is a DataFrame that contains information about the weather in the city of Seattle in the different months of the year. The "MONTH" column contains the three-letter names of the months of the year. The "monthly average normal temperature" column contains the temperatures in these months, in Fahrenheit degrees, averaged over a ten-year period.

## **5. Adding data to axes**

02:50 - 03:41

To add the data to the Axes, we call a plotting command. The plotting commands are methods of the Axes object. For example, here we call the method called `plot` with the month column as the first argument and the temperature column as the second argument. Finally, we call the `plt.show` function to show the effect of the plotting command. This adds a line to the plot. The horizontal dimension of the plot represents the months according to their order and the height of the line at each month represents the average temperature. The trends in the data are now much clearer than they were just by reading off the temperatures from the table.

## **6. Adding more data**

03:41 - 03:59

If you want, you can add more data to the plot. For example, we also have a table that stores data about the average temperatures in the city of Austin, Texas. We add these data to the axes by calling the `plot` method again.

## **7. Putting it all together**

03:59 - 04:21

Here is what all of the code to create this figure would then look like. First, we create the Figure and the Axes objects. We call the Axes method `plot` to add first the Seattle temperatures, and then the Austin temperatures to the Axes. Finally, we ask Matplotlib to show us the figure.

## **8. Practice making a figure!**

04:21 - 04:33

Now it's your turn. In the exercises, you will practice making a figure and axes and adding data into them.