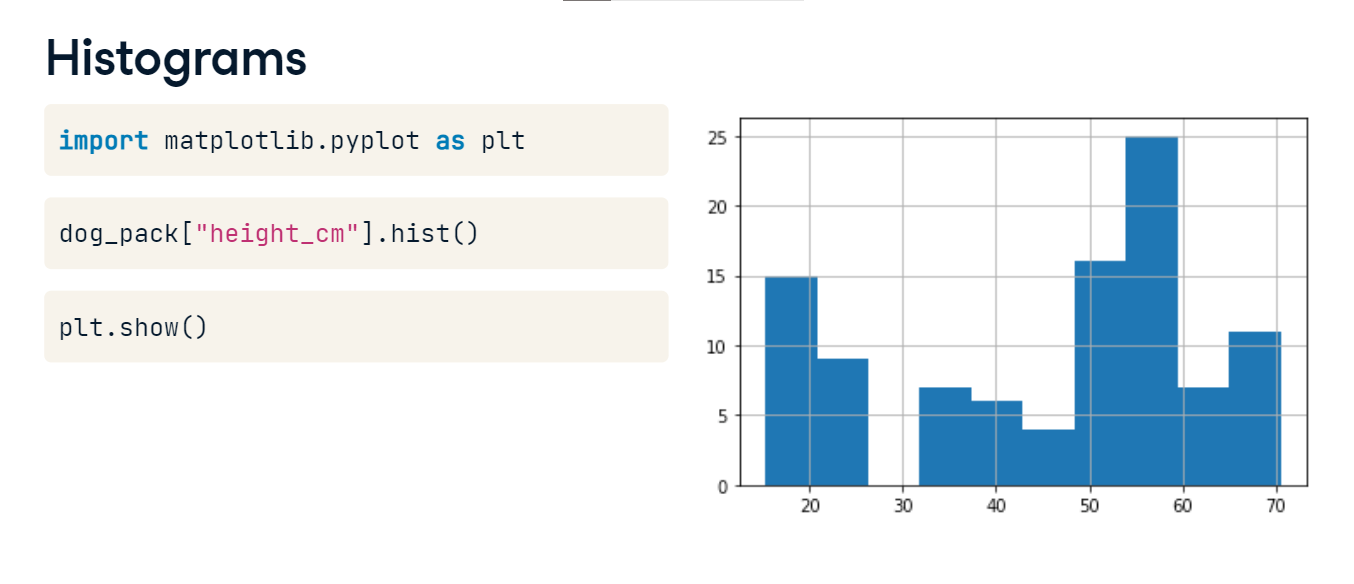
**Visualizing your data**

Plots are a powerful way to share the insights you've gained from your data. In this lesson, we'll use a bigger dataset of dogs, called dog\_pack, to make visualization easier.

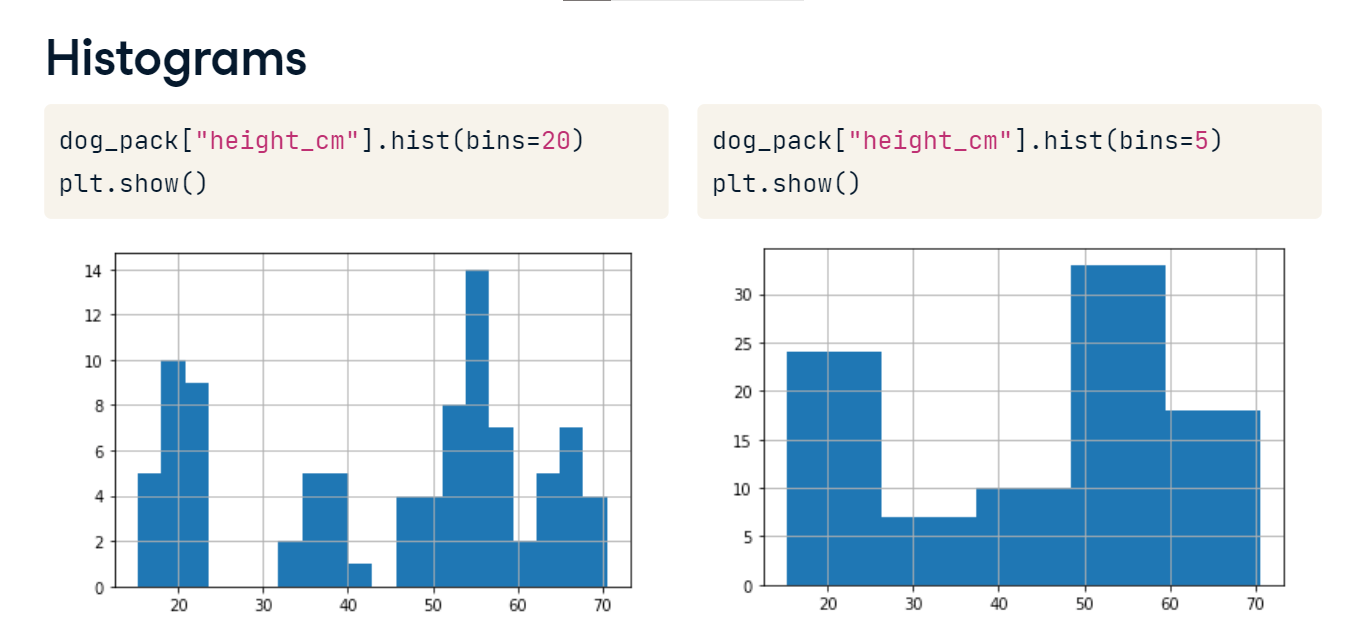
**Histograms**

Remember when we talked about matplotlib at the beginning of the course? We'll need to import matplotlib-dot-pyplot as plt in order to display our visualizations. Just like pd is the standard alias for pandas, plt is the standard alias for matplotlib-dot-pyplot. Let's create a histogram, which shows the distribution of a numeric variable. We can create a histogram of the height variable by selecting the column and calling dot-hist. In order to show the plot, we need to call plt-dot-show. The x-axis represents the heights of the dogs, and the y-axis represents the number of dogs in each height range. By grouping observations into ranges, the histogram allows us to see that there are a lot of dogs around 50 to 60 centimeters tall.



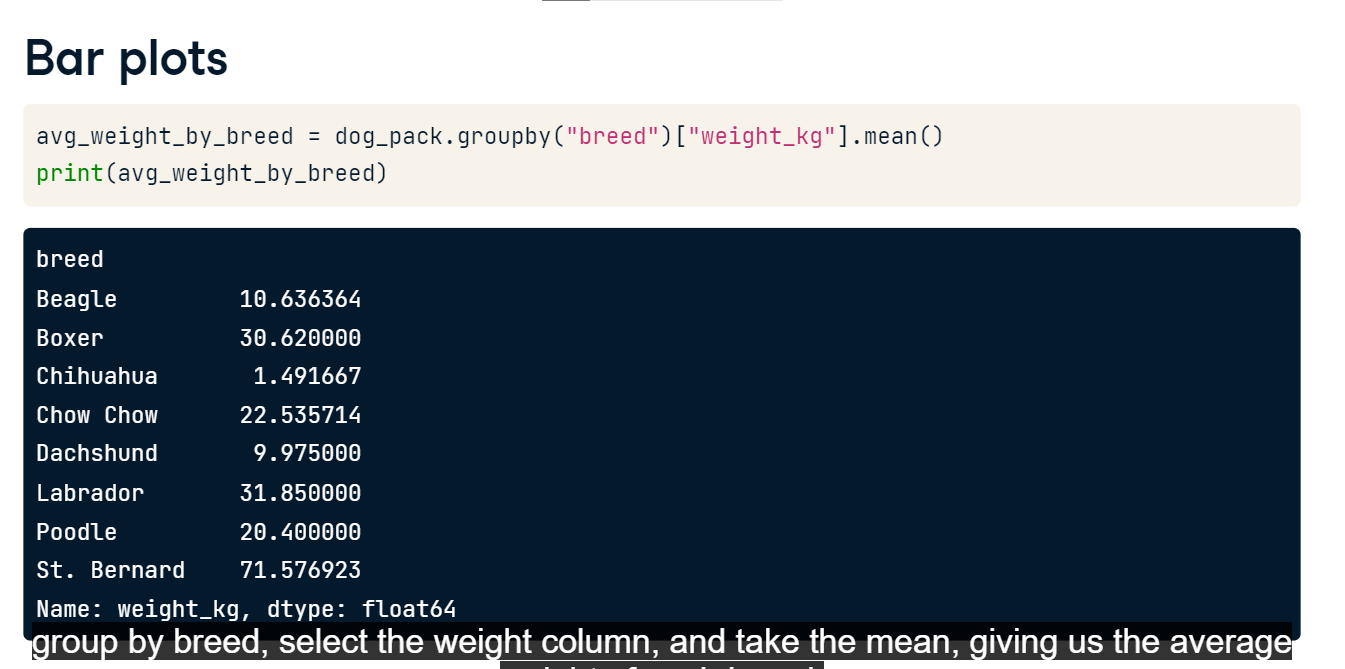
**Histograms**

We can adjust the number of bars, or bins, using the "bins" argument. Increasing or decreasing this can give us a better idea of what the distribution looks like.



**Bar plots**

Bar plots can reveal relationships between a categorical variable and a numeric variable, like breed and weight. To compute the average weight of each breed, we group by breed, select the weight column, and take the mean, giving us the average weight of each breed.



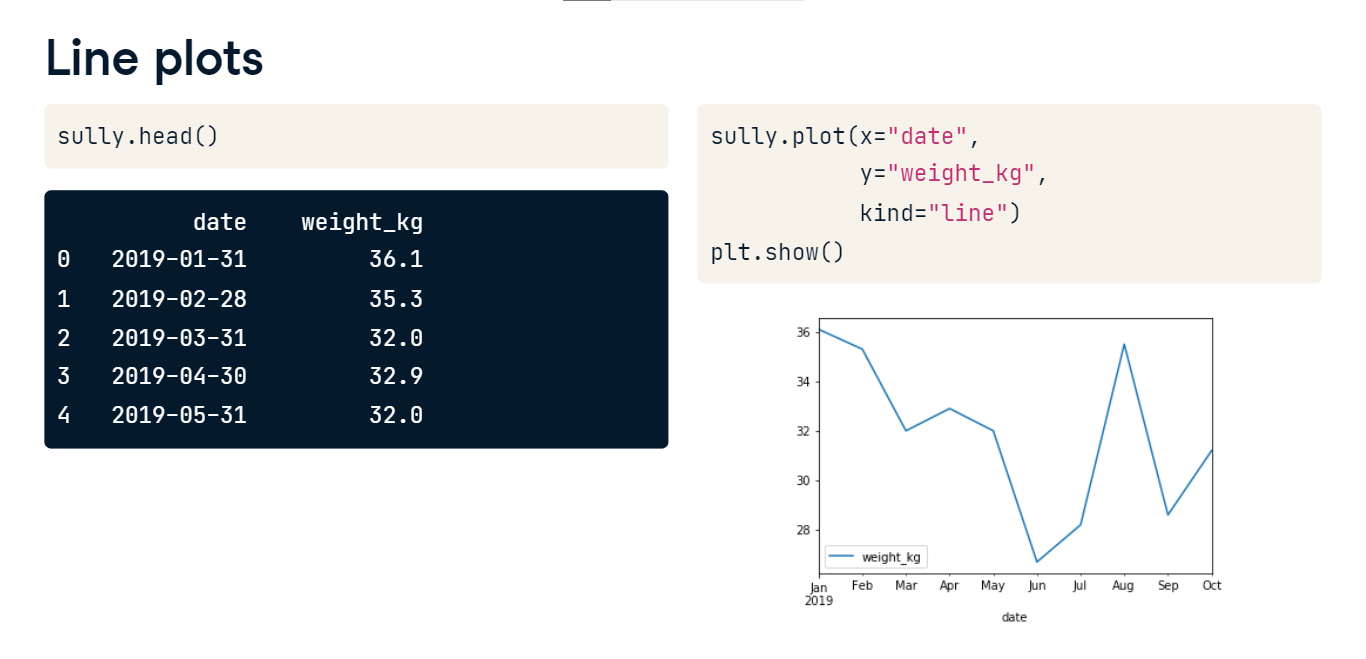
**Bar plots**

Now we can create a bar plot from the mean weights using the plot method, setting "kind" equal to "bar." Finally, we call plt-dot-show. To add a title to our plot, we can use the title argument of the plot method. It looks like Saint Bernards are the heaviest breed on average! Woof!



**Line plots**

Line plots are great for visualizing changes in numeric variables over time. Lucky for us, a Labrador named Sully has been weighed by his owner every month - let's see how his weight has changed over the year. We can use the plot method again, but this time, we pass in three arguments: date as x, weight as y, and "kind" equals "line." Sully's weight has fluctuated quite a bit over the year!



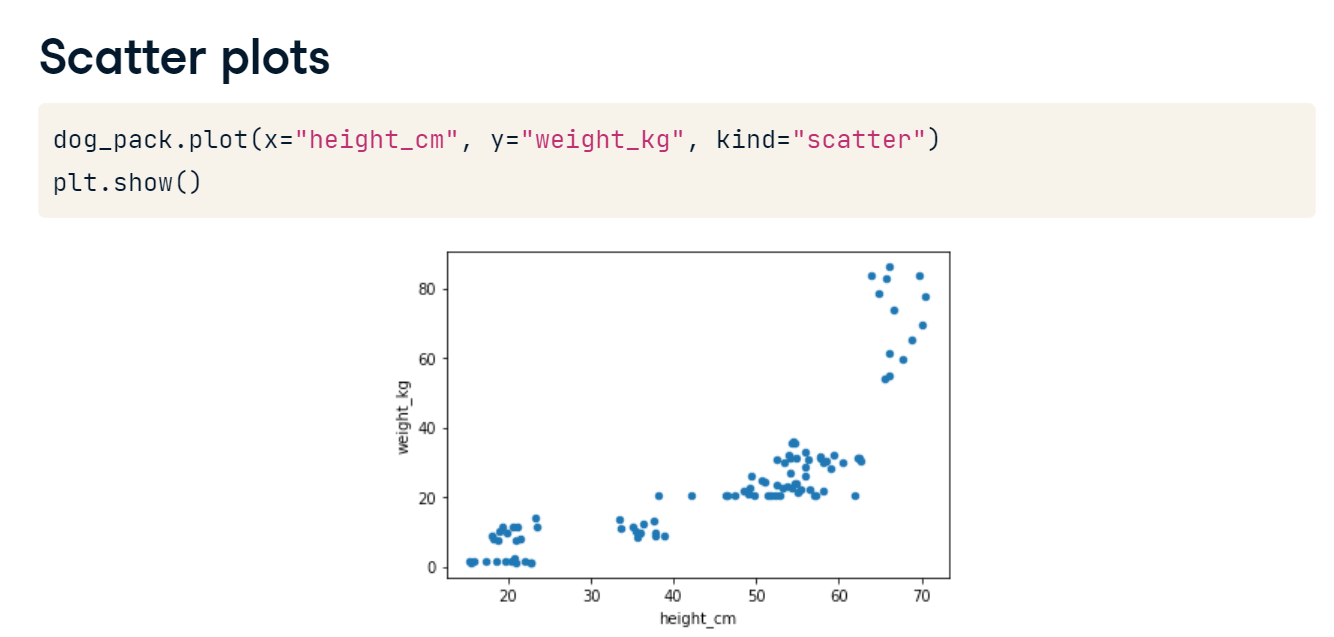
**Rotating axis labels**

We may want to rotate the x-axis labels to make the text easier to read. This can be done by passing an angle in degrees with the "rot" argument. Here, we rotate the labels by 45 degrees.



**Scatter plots**

Scatter plots are great for visualizing relationships between two numeric variables. To plot each dog's height versus their weight, we call the plot method with x equal to height\_cm, y equal to weight\_kg, and "kind" equal to "scatter." From our plot, it looks like taller dogs tend to weigh more.

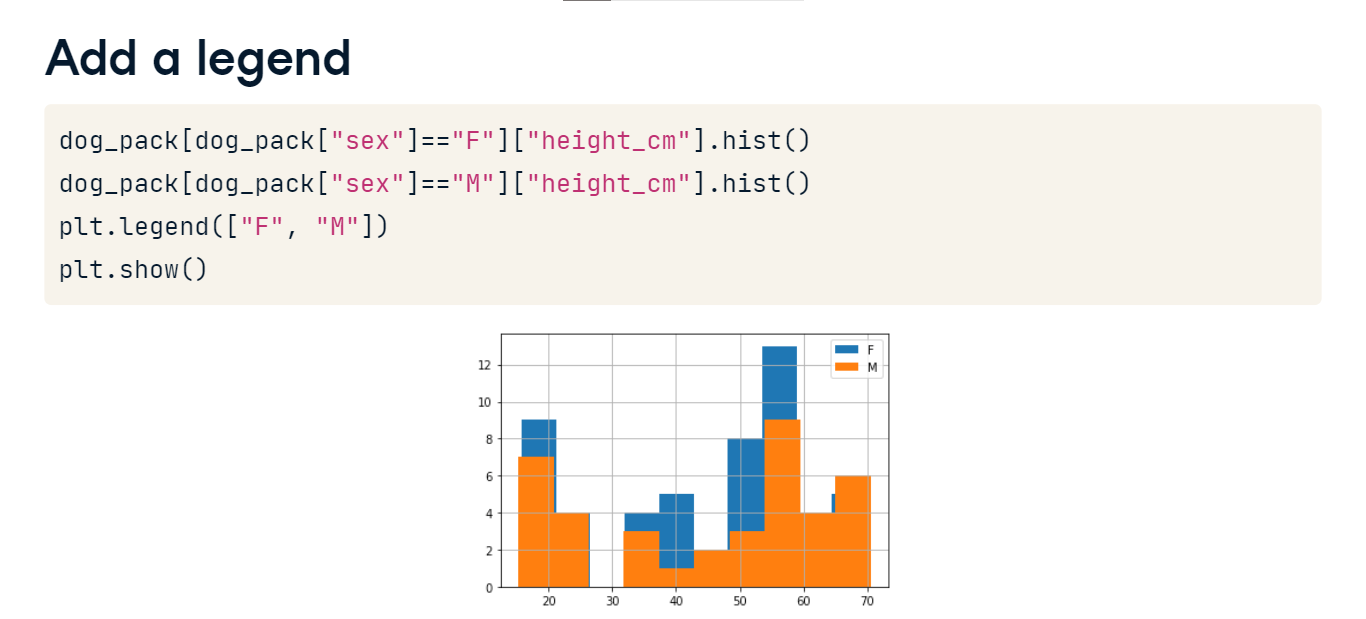


**Layering plots**

Plots can also be layered on top of one another. For example, we can create a histogram of female dogs' heights, and put a histogram of male dogs' heights on top, then call show. However, we can't tell which color represents which sex.

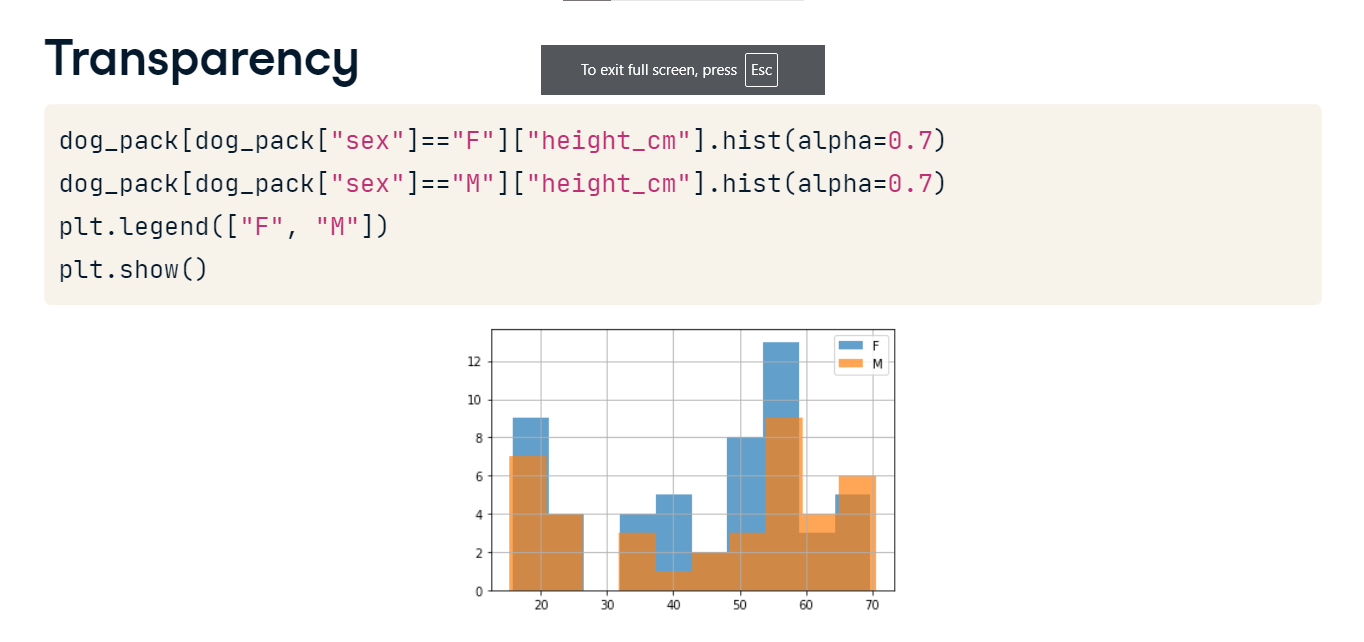
**Add a legend**

We can use plt-dot-legend, passing in a list of labels, and then call show. Now we know which color is which, but we can't see what's going on behind the orange histogram.

****

**Transparency**

Let's fix this problem by making the histograms translucent. We can use hist's alpha argument, which takes a number. 0 means completely transparent that is, invisible, and 1 means completely opaque.



**Avocados**

In this chapter, you'll be working with a dataset that contains weekly US avocado sales data, broken down by avocado size, and whether or not the avocados were organic.

