**1. Sampling and point estimates**

Hi! Welcome to the course! I’m James, and I’ll be your host as we delve into the world of sampling data with Python. To start, let’s look at what sampling is and why it might be useful.

**2. Estimating the population of France**

Let's consider the problem of counting how many people live in France. The standard approach is to take a census. This means contacting every household and asking how many people live there.

**3. There are lots of people in France**

Since there are millions of people in France, this is a really expensive process. Even with modern data collection technology, most countries will only conduct a census every five or ten years due to the cost.

**4. Sampling households**

In 1786, Pierre-Simon Laplace realized you could estimate the population with less effort. Rather than asking every household who lived there, he asked a small number of households and used statistics to estimate the number of people in the whole population. This technique of working with a subset of the whole population is called sampling.

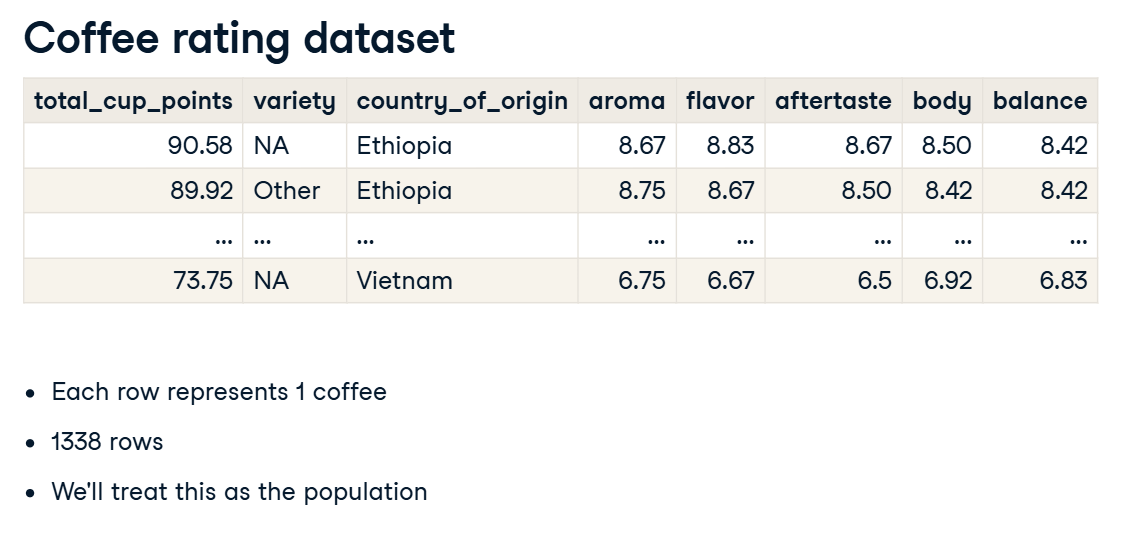
**5. Population vs. sample**

Two definitions are important for this course. The population is the complete set of data that we are interested in. The previous example involved the literal population of France, but in statistics, it doesn't have to refer to people. One thing to bear in mind is that there is usually no equivalent of the census, so typically, we won't know what the whole population is like - more on this in a moment. The sample is the subset of data that we are working with.

**6. Coffee rating dataset**

01:39 - 02:26

Here's a dataset of professional ratings of coffees. Each row corresponds to one coffee, and there are thirteen hundred and thirty-eight rows in the dataset. The coffee is given a score from zero to one hundred, which is stored in the total\_cup\_points column. Other columns contain contextual information like the variety and country of origin and scores between zero and ten for attributes of the coffee such as aroma and body. These scores are averaged across all the reviewers for that particular coffee. It doesn't contain every coffee in the world, so we don't know exactly what the population of coffees is. However, there are enough here that we can think of it as our population of interest.



**7. Points vs. flavor: population**

Let's consider the relationship between cup points and flavor by selecting those two columns. This dataset contains all thirteen hundred and thirty-eight rows from the original dataset.

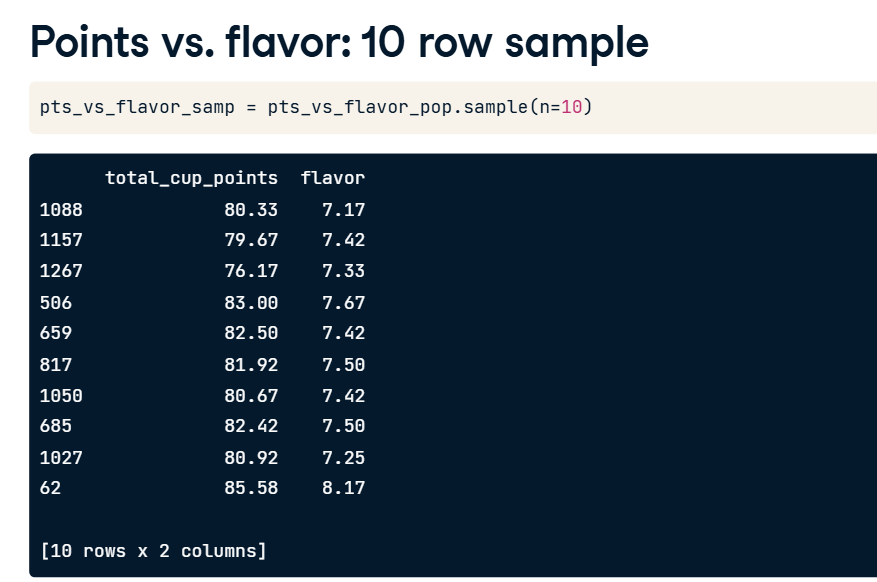
**8. Points vs. flavor: 10 row sample**

The pandas dot-sample method returns a random subset of rows. Setting n to ten means ten random rows are returned. By default, rows from the original dataset can't appear in the sample dataset multiple times, so we are guaranteed to have ten unique rows in our sample.



**9. Python sampling for Series**

The dot-sample method also works on pandas Series. Here, using square-bracket subsetting retrieves the total\_cup\_points column as a Series, and the n argument specifies how many random values to return.



**10. Population parameters & point estimates**

03:14 - 03:45

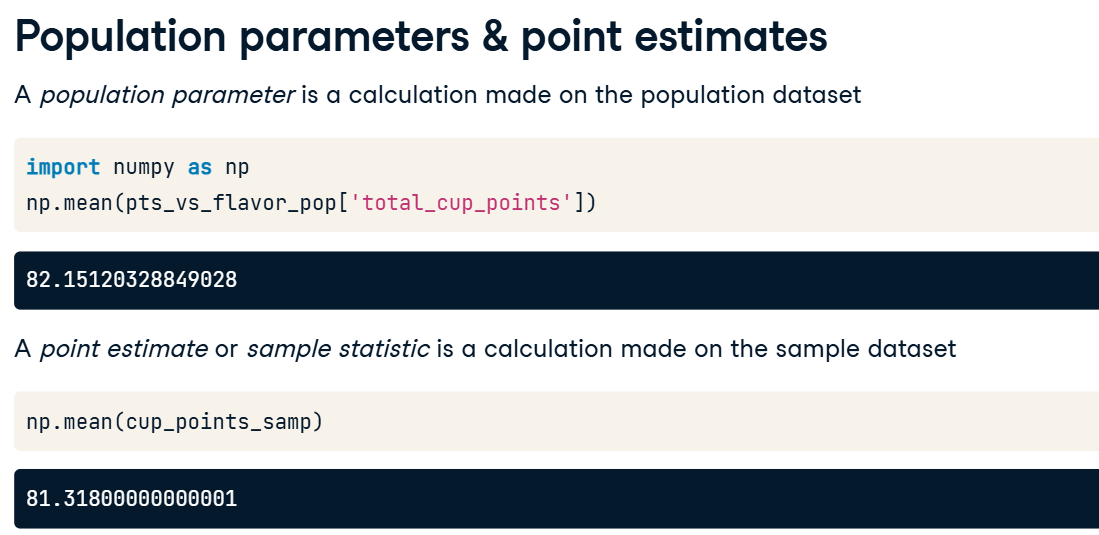
A population parameter is a calculation made on the population dataset. We aren't limited to counting values either; here, we calculate the mean of the cup points using NumPy. By contrast, a point estimate, or sample statistic, is a calculation based on the sample dataset. Here, the mean of the total cup points is calculated on the sample. Notice that the means are very similar but not identical.



**11. Point estimates with pandas**

03:45 - 03:54

Working with pandas can be easier than working with NumPy. These mean calculations can be performed using the dot-mean pandas method.



**1. Convenience sampling**

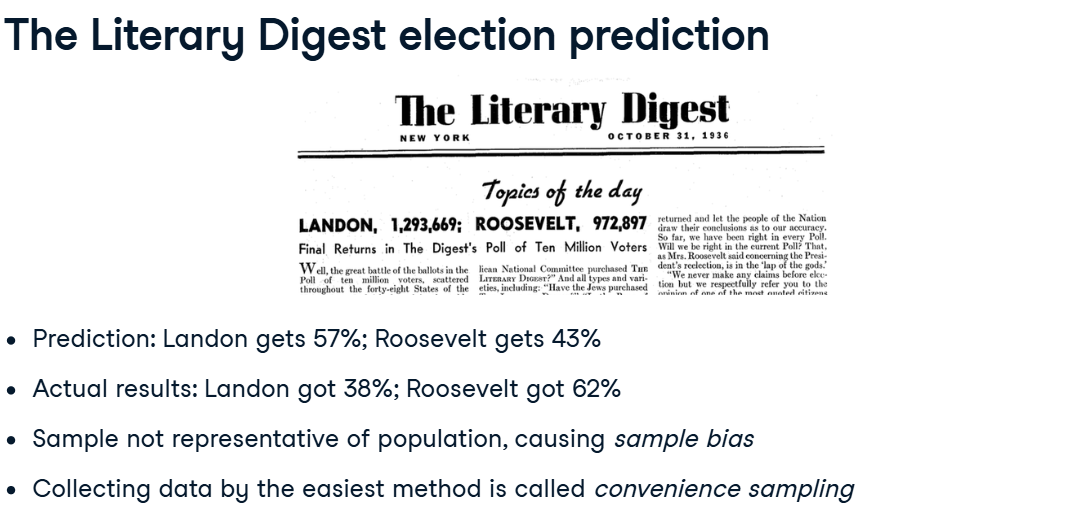
00:00 - 00:11

The point estimates you calculated in the previous exercises were very close to the population parameters that they were based on, but is this always the case?

**2. The Literary Digest election prediction**

00:11 - 01:34

In 1936, a newspaper called The Literary Digest ran an extensive poll to try to predict the next US presidential election. They phoned ten million voters and had over two million responses. About one-point-three million people said they would vote for Landon, and just under one million people said they would vote for Roosevelt. That is, Landon was predicted to get fifty-seven percent of the vote, and Roosevelt was predicted to get forty-three percent of the vote. Since the sample size was so large, it was presumed that this poll would be very accurate. However, in the election, Roosevelt won by a landslide with sixty-two percent of the vote. So what went wrong? Well, in 1936, telephones were a luxury, so the only people who had been contacted by The Literary Digest were relatively rich. The sample of voters was not representative of the whole population of voters, and so the poll suffered from sample bias. The data was collected by the easiest method, in this case, telephoning people. This is called convenience sampling and is often prone to sample bias. Before sampling, we need to think about our data collection process to avoid biased results.



**3. Finding the mean age of French people**

01:34 - 01:57

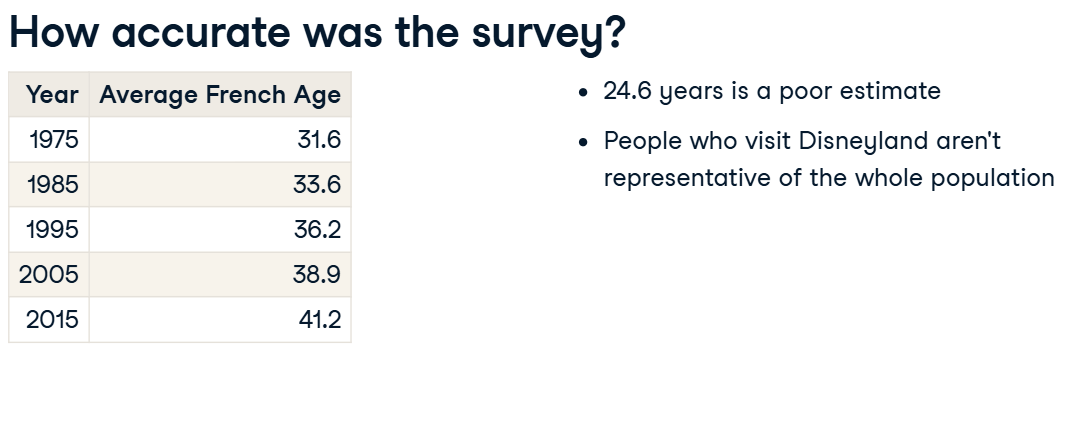
Let's look at another example. While on vacation at Disneyland Paris, you start wondering about the mean age of French people. To get an answer, you ask ten people stood nearby about their ages. Their mean age is twenty-four-point-six years old. Do you think this will be a good estimate of the mean age of all French citizens?

1. 1 Image by Sean MacEntee

**4. How accurate was the survey?**

01:57 - 02:32

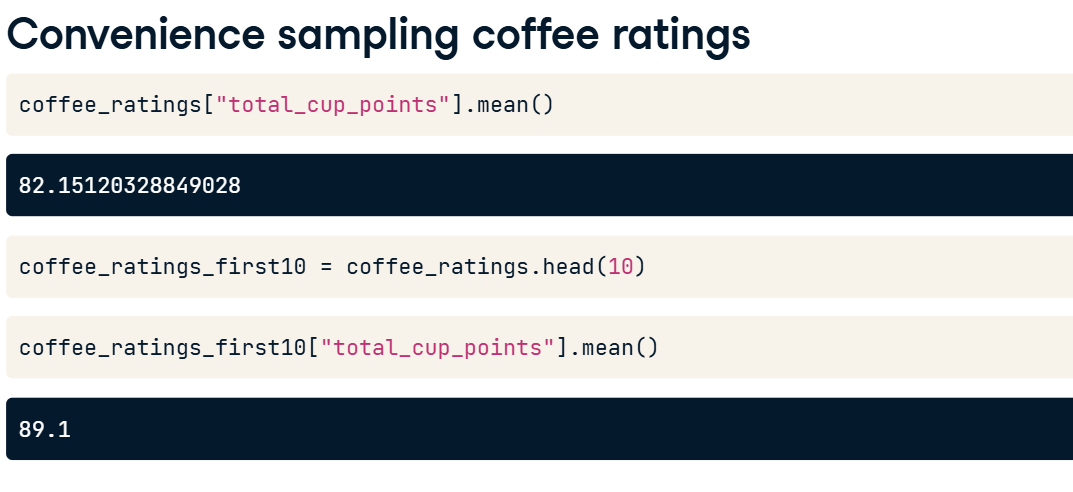
On the left, you can see mean ages taken from the French census. Notice that the population has been gradually getting older as birth rates decrease and life expectancy increases. In 2015, the mean age was over forty, so our estimate of twenty-four-point-six is way off. The problem is that the family-friendly fun at Disneyland means that the sample ages weren't representative of the general population. There are generally more eight-year-olds than eighty-year-olds riding rollercoasters.



**5. Convenience sampling coffee ratings**

02:32 - 03:10

Let's return to the coffee ratings dataset and look at the mean cup points population parameter. The mean is about eighty-two. One form of convenience sampling would be to take the first ten rows, rather than the random rows we saw in the previous video. We can take the first 10 rows with the pandas head method. The mean cup points from this sample is higher at eighty-nine. The discrepancy suggests that coffees with higher cup points appear near the start of the dataset. Again, the convenience sample isn't representative of the whole population.



**6. Visualizing selection bias**

03:10 - 03:44

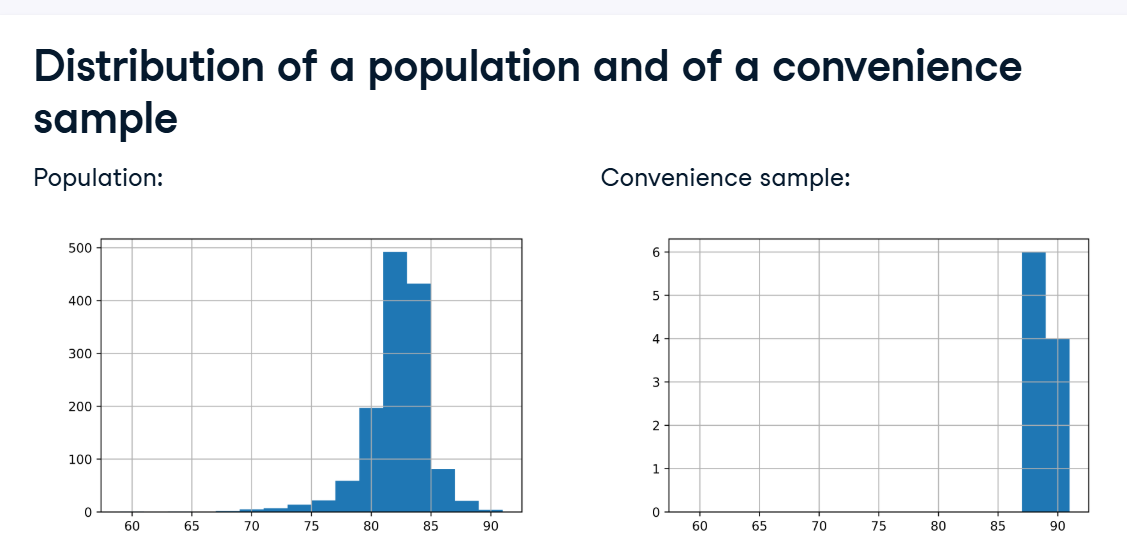
Histograms are a great way to visualize the selection bias. We can create a histogram of the total cup points from the population, which contains values ranging from around 59 to around 91. The numpy-dot-arange function can be used to create bins of width 2 from 59 to 91. Recall that the stop value in numpy-dot-arange is exclusive, so we specify 93, not 91. Here's the same code to generate a histogram for the convenience sample.



**7. Distribution of a population and of a convenience sample**

03:44 - 03:54

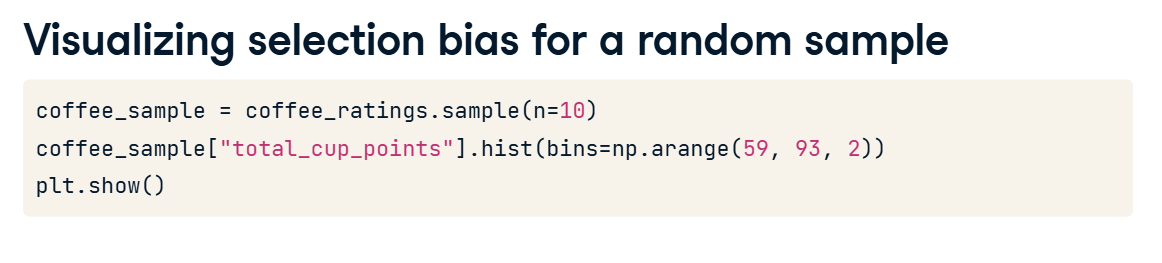
Comparing the two histograms, it is clear that the distribution of the sample is not the same as the population: all of the sample values are on the right-hand side of the plot.



**8. Visualizing selection bias for a random sample**

03:54 - 04:02

This time, we'll compare the total\_cup\_points distribution of the population with a random sample of 10 coffees.



**9. Distribution of a population and of a simple random sample**

04:02 - 04:09

Notice how the shape of the distributions is more closely aligned when random sampling is used.

**Pseudo-random number generation**

You previously saw how to use a random sample to get results similar to those in the population. But how does a computer actually do this random sampling?

**2. What does random mean?**

There are several meanings of random in English. This definition from Oxford Languages is the most interesting for us. If we want to choose data points at random from a population, we shouldn't be able to predict which data points would be selected ahead of time in some systematic way.

1. 1 Oxford Languages

**3. True random numbers**

To generate truly random numbers, we typically have to use a physical process like flipping coins or rolling dice. The Hotbits service generates numbers from radioactive decay, and RANDOM-dot-ORG generates numbers from atmospheric noise, which are radio signals generated by lightning. Unfortunately, these processes are fairly slow and expensive for generating random numbers.

1. 1 https://www.fourmilab.ch/hotbits
2. 2 https://www.random.org

**4. Pseudo-random number generation**

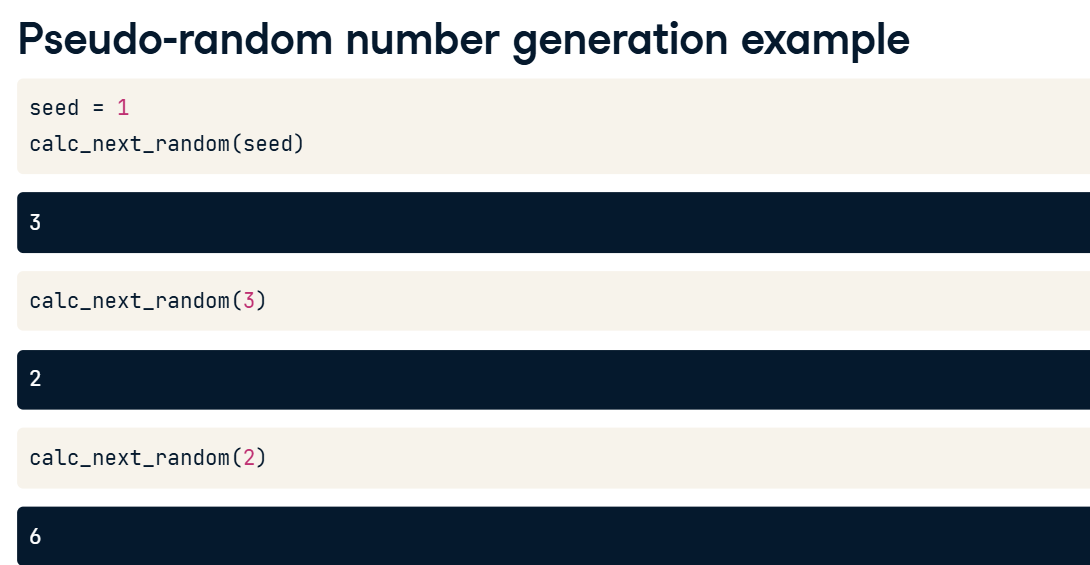
00:56 - 01:32

For most use cases, pseudo-random number generation is better since it is cheap and fast. Pseudo-random means that although each value appears to be random, it is actually calculated from the previous random number. Since you have to start the calculations somewhere, the first random number is calculated from what is known as a seed value. The word random is in quotes to emphasize that this process isn't really random. If we start from a particular seed value, all future numbers will be the same.

**5. Pseudo-random number generation example**

01:32 - 02:10

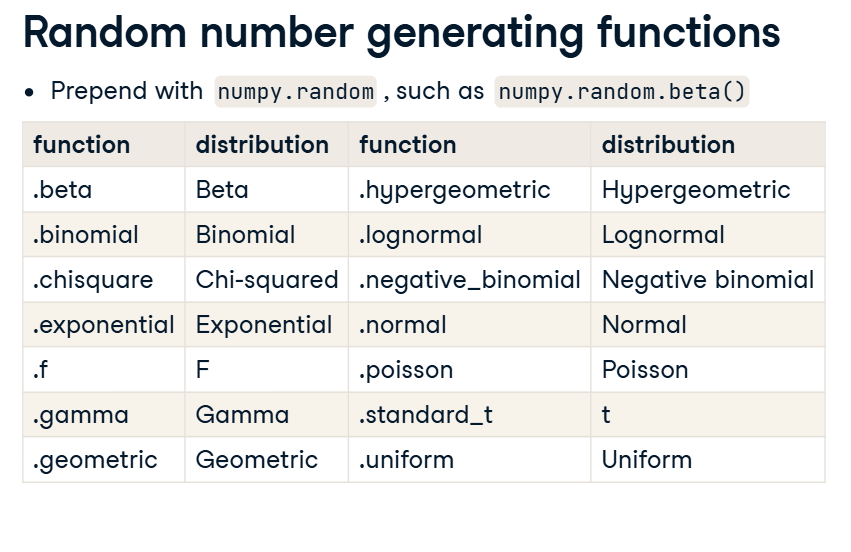
For example, suppose we have a function to generate pseudo-random values called calc\_next\_random. To begin, we pick a seed number, in this case, one. calc\_next\_random does some calculations and returns three. We then feed three into calc\_next\_random, and it does the same set of calculations and returns two. And if we can keep feeding in the last number, it will return something apparently random. Although the process is deterministic, the trick to a random number generator is to make it look like the values are random.



**6. Random number generating functions**

02:10 - 02:33

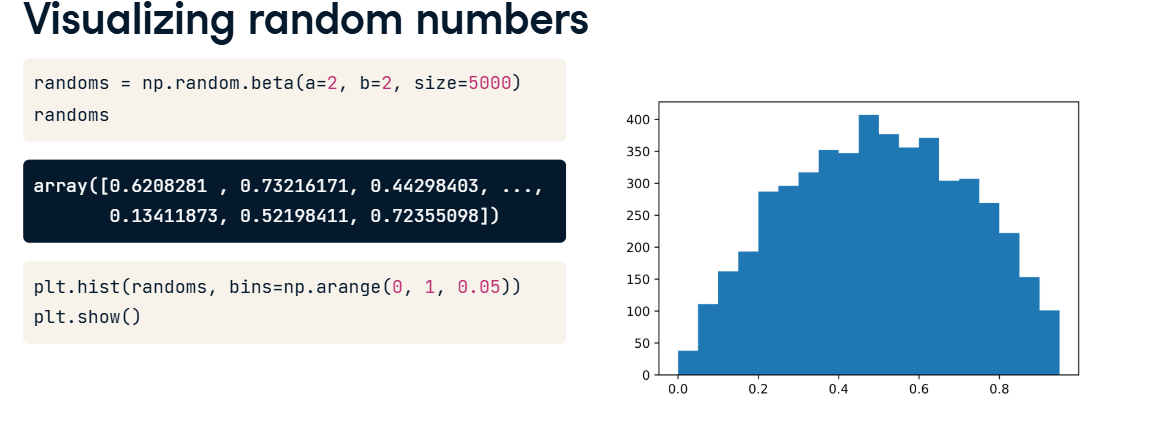
NumPy has many functions for generating random numbers from statistical distributions. To use each of these, make sure to prepend each function name with numpy-dot-random or np-dot-random. Some of them, like dot-uniform and dot-normal, may be familiar. Others have more niche applications.



**7. Visualizing random numbers**

02:33 - 03:10

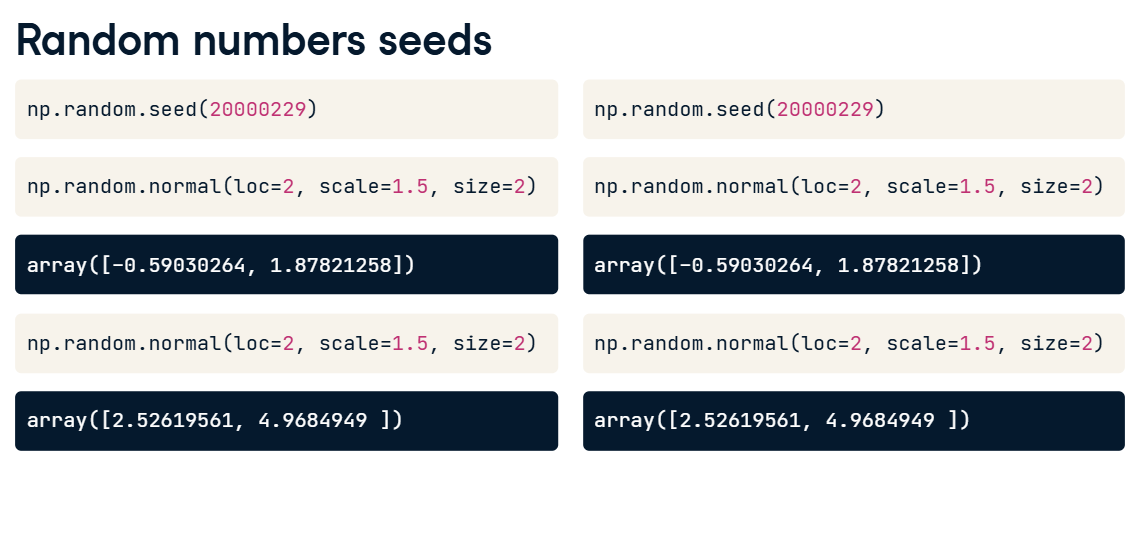
Let's generate some pseudo-random numbers. The first arguments to each random number function specify distribution parameters. The size argument specifies how many numbers to generate, in this case, five thousand. We've chosen the beta distribution, and its parameters are named a and b. These random numbers come from a continuous distribution, so a great way to visualize them is with a histogram. Here, because the numbers were generated from the beta distribution, all the values are between zero and one.



**8. Random numbers seeds**

03:10 - 04:00

To set a random seed with NumPy, we use the dot-random-dot-seed method. random-dot-seed takes an integer for the seed number, which can be any number you like. dot-normal generates pseudo-random numbers from the normal distribution. The loc and scale arguments set the mean and standard deviation of the distribution, and the size argument determines how many random numbers from that distribution will be returned. If we call dot-normal a second time, we get two different random numbers. If we reset the seed by calling random-dot-seed with the same seed number, then call dot-normal again, we get the same numbers as before. This makes our code reproducible.



**9. Using a different seed**

04:00 - 04:07

Now let's try a different seed. This time, calling dot-normal generates different numbers.