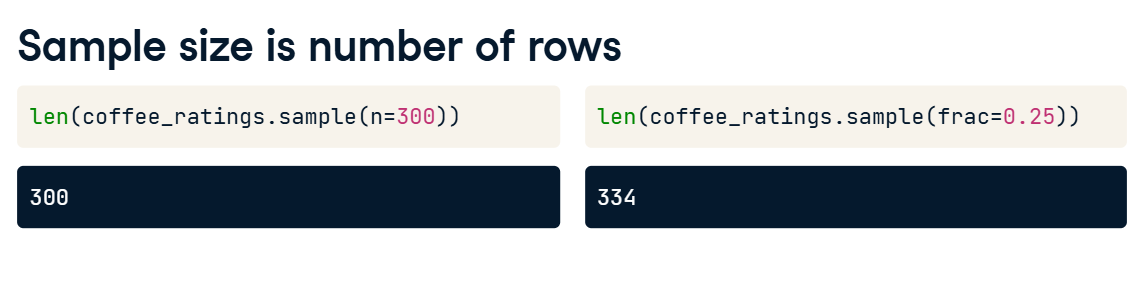
**Relative error of point estimates**

Let's see how the size of the sample affects the accuracy of the point estimates we calculate.

**2. Sample size is number of rows**

00:06 - 00:26

The sample size, calculated here with the len function, is the number of observations, that is, the number of rows in the sample. That's true whichever method we use to create the sample. We'll stick to looking at simple random sampling since it works well in most cases and it's easier to reason about.



**3. Various sample sizes**

00:26 - 01:09

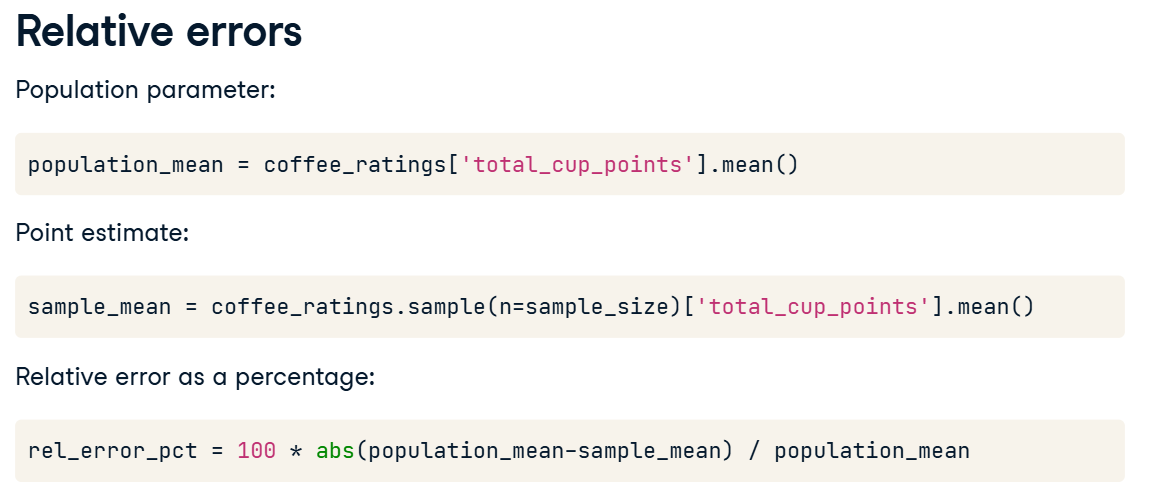
Let's calculate a population parameter, the mean cup points of the coffees. It's around eighty-two-point-one-five. This is our gold standard to compare against. If we take a sample size of ten, the point estimate of this parameter is wrong by about point-eight-eight. Increasing the sample size to one hundred gets us closer; the estimate is only wrong by about point-three-four. Increasing the sample size further to one thousand brings the estimate to about point-zero-three away from the population parameter. In general, larger sample sizes will give us more accurate results.



**4. Relative errors**

01:09 - 01:45

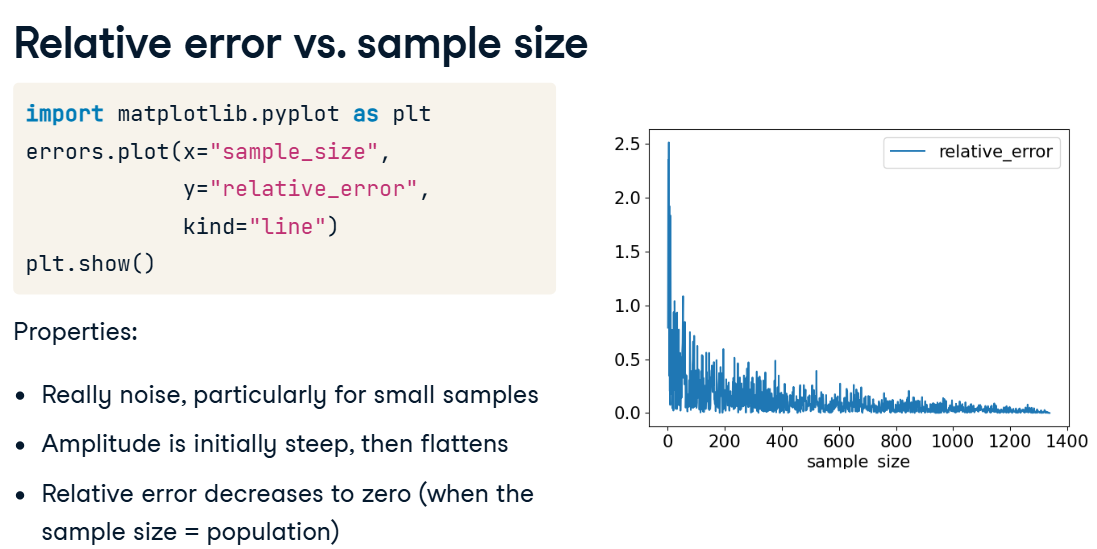
For any of these sample sizes, we want to compare the population mean to the sample mean. This is the same code we just saw, but with the numerical sample size replaced with a variable named sample\_size. The most common metric for assessing the difference between the population and a sample mean is the relative error. The relative error is the absolute difference between the two numbers; that is, we ignore any minus signs, divided by the population mean. Here, we also multiply by one hundred to make it a percentage.



**5. Relative error vs. sample size**

01:45 - 02:47

Here's a line plot of relative error versus sample size. We see that the relative error decreases as the sample size increases, and beyond that, the plot has other important properties. Firstly, the blue line is really noisy, particularly for small sample sizes. If our sample size is small, the sample mean we calculate can be wildly different by adding one or two more random rows to the sample. Secondly, the amplitude of the line is quite steep, to begin with. When we have a small sample size, adding just a few more samples can give us much better accuracy. Further to the right of the plot, the line is less steep. If we already have a large sample size, adding a few more rows to the sample doesn't bring as much benefit. Finally, at the far right of the plot, where the sample size is the whole population, the relative error decreases to zero.



**Creating a sampling distribution**

We just saw how point estimates like the sample mean will vary depending on which rows end up in the sample.

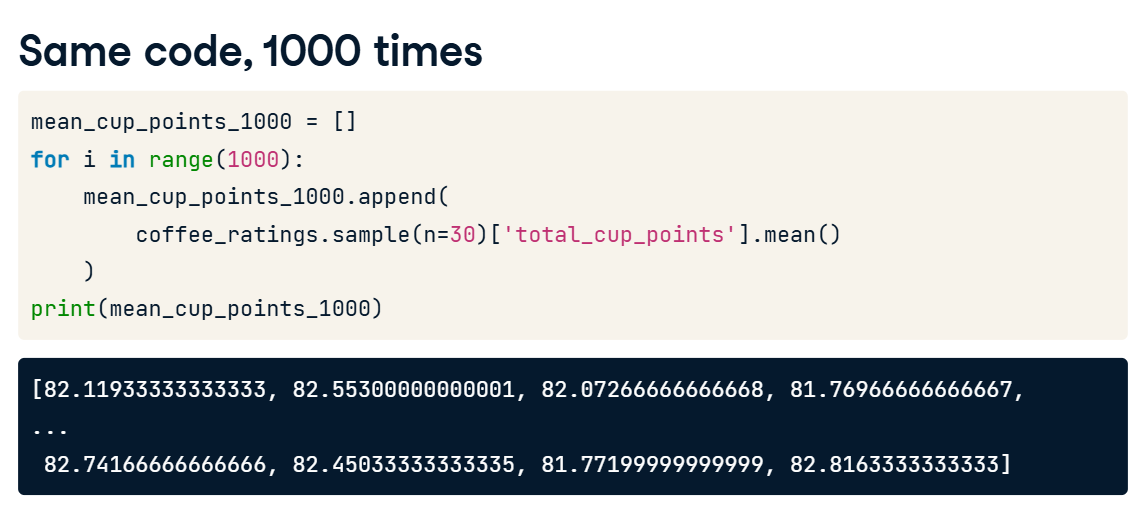
**2. Same code, different answer**

For example, this same code to calculate the mean cup points from a simple random sample of thirty coffees gives a slightly different answer each time. Let's try to visualize and quantify this variation.



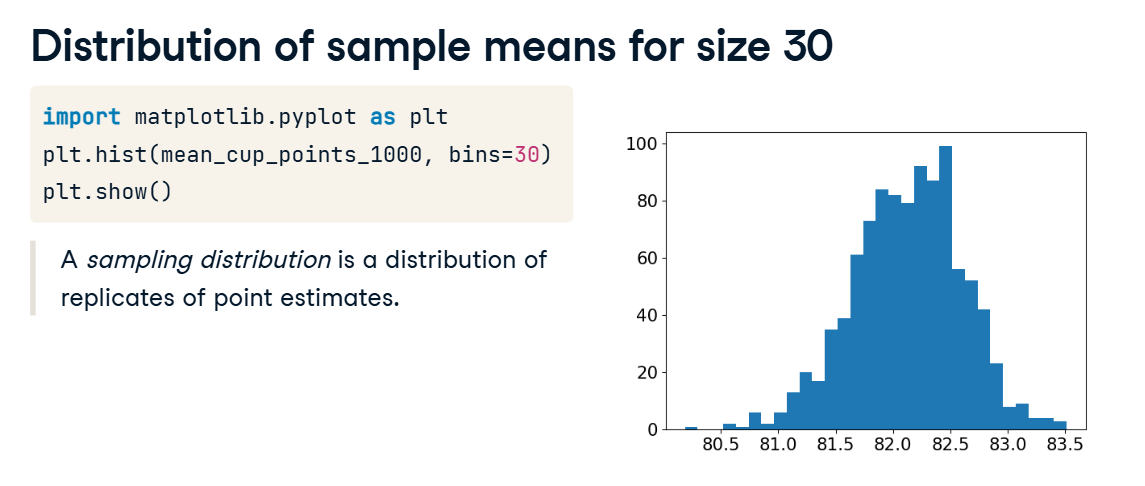
**3. Same code, 1000 times**

A for loop lets us run the same code many times. It's especially useful for situations like this where the result contains some randomness. We start by creating an empty list to store the means. Then, we set up the for loop to repeatedly sample 30 coffees from coffee\_ratings a total of 1000 times, calculating the mean cup points each time. After each calculation, we append the result, also called a replicate, to the list. Each time the code is run, we get one sample mean, so running the code a thousand times generates a list of one thousand sample means.



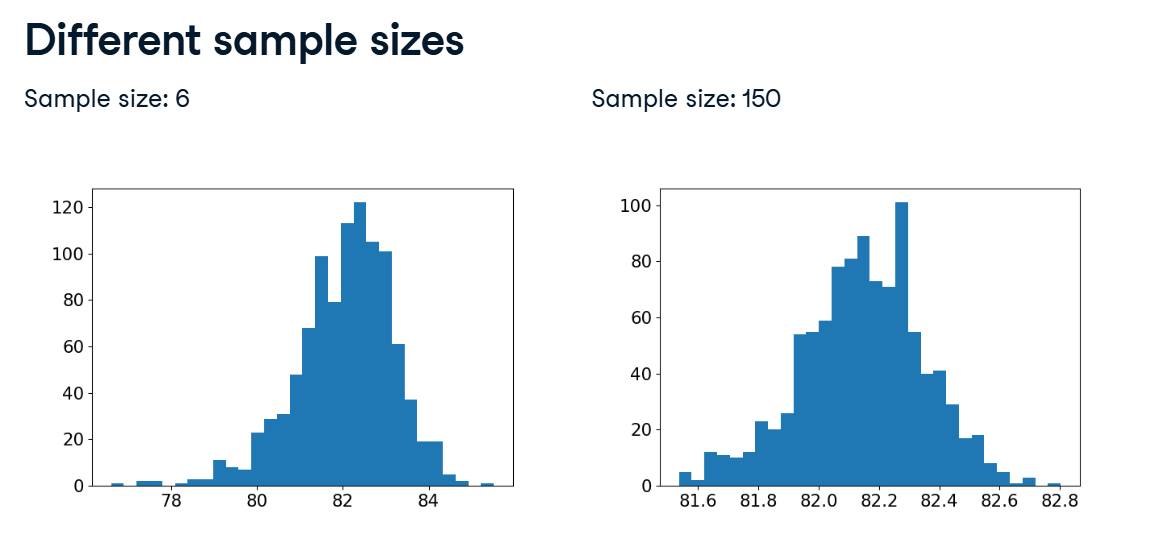
**4. Distribution of sample means for size 30**

The one thousand sample means form a distribution of sample means. To visualize a distribution, the best plot is often a histogram. Here we can see that most of the results lie between eighty-one and eighty-three, and they roughly follow a bell-shaped curve, like a normal distribution. There's an important piece of jargon we need to know here. A distribution of replicates of sample means, or other point estimates, is known as a sampling distribution.



**5. Different sample sizes**

Here are histograms from running the same code again with different sample sizes. When we decrease the original sample size of thirty to six, we can see from the x-values that the range of the results is broader. The bulk of the results now lie between eighty and eighty-four. On the other hand, increasing the sample size to one hundred and fifty results in a much narrower range. Now most of the results are between eighty-one-point-eight and eighty-two-point-six. As we saw previously, bigger sample sizes give us more accurate results. By replicating the sampling many times, as we've done here, we can quantify that accuracy.



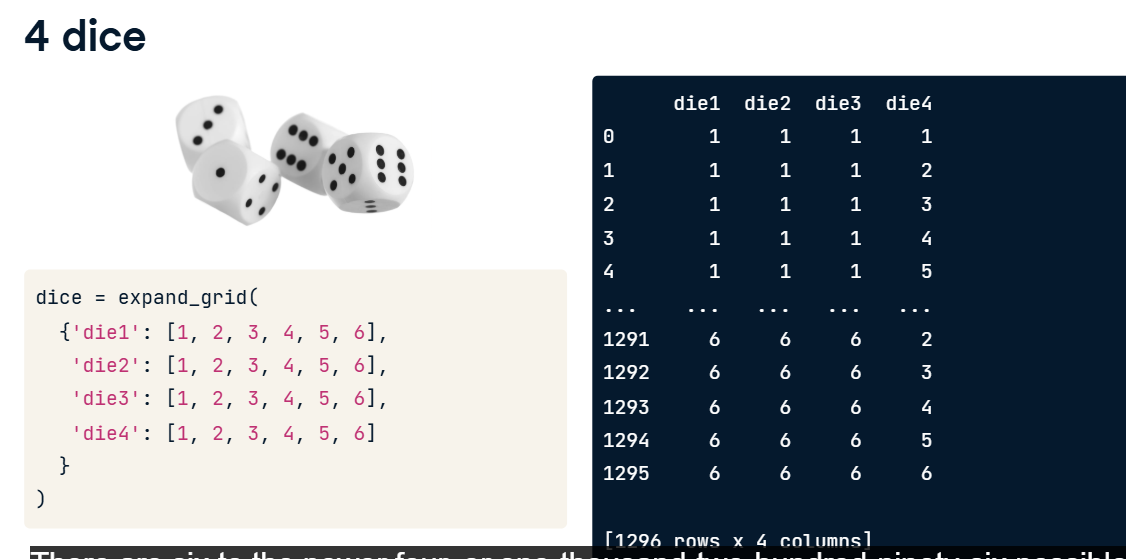
**1. Approximate sampling distributions**

In the last exercise, we saw that while increasing the number of replicates didn't affect the relative error of the sample means; it did result in a more consistent shape to the distribution.

**2. 4 dice**

00:12 - 00:34

Let's consider the case of four six-sided dice rolls. We can generate all possible combinations of rolls using the expand\_grid function, which is defined in the pandas documentation, and uses the itertools package. There are six to the power four, or one-thousand-two-hundred-ninety-six possible dice roll combinations.



**3. Mean roll**

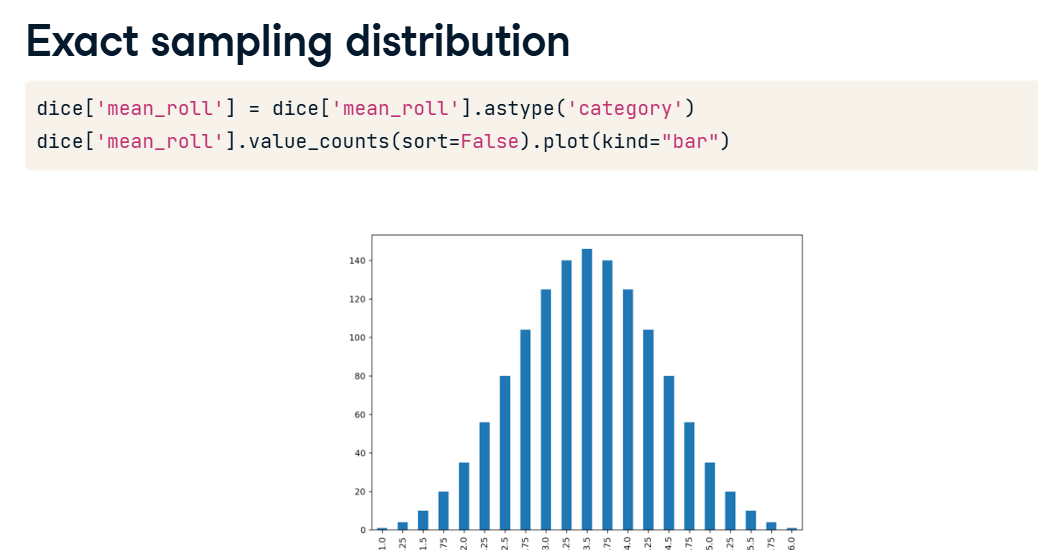
00:34 - 00:48

Let's consider the mean of the four rolls by adding a column to our DataFrame called mean\_roll. mean\_roll ranges from 1, when four ones are rolled, to 6, when four sixes are rolled.

**4. Exact sampling distribution**

00:48 - 01:33

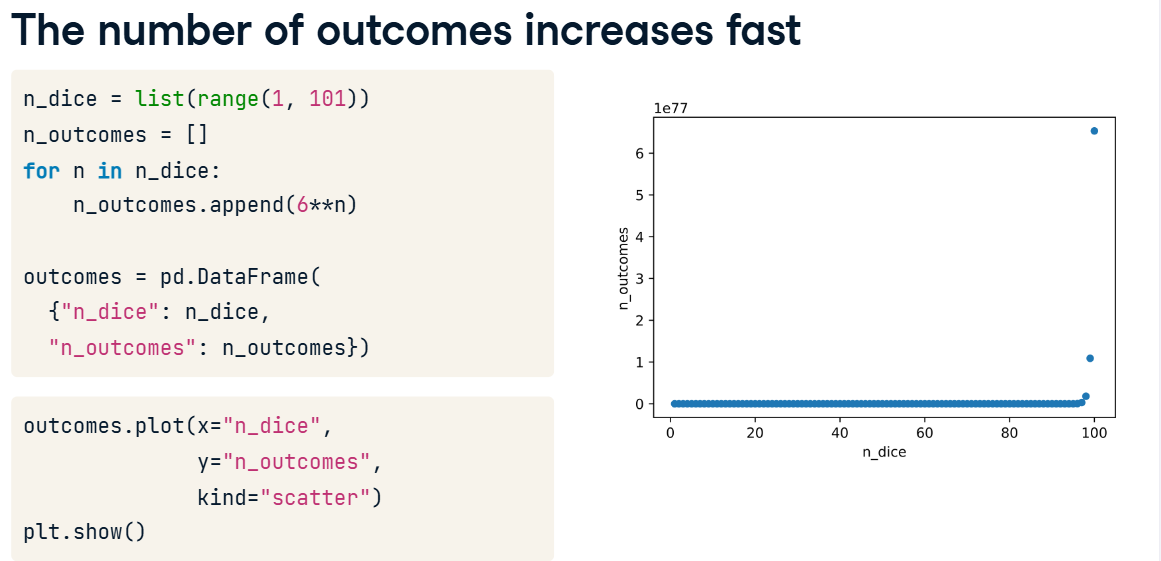
Since the mean roll takes discrete values instead of continuous values, the best way to see the distribution of mean\_roll is to draw a bar plot. First, we convert mean\_roll to a categorical by setting its type to category. We are interested in the counts of each value, so we use dot-value\_counts, passing the sort equals False argument. This ensures the x-axis ranges from one to six instead of sorting the bars by frequency. Chaining dot-plot to value\_counts, and setting kind to "bar", produces a bar plot of the mean roll distribution. This is the exact sampling distribution of the mean roll because it contains every single combination of die rolls.



**5. The number of outcomes increases fast**

01:33 - 02:19

If we increase the number of dice in our scenario, the number of possible outcomes increases by a factor of six each time. These values can be shown by creating a DataFrame with two columns: n\_dice, ranging from 1 to 100, and n\_outcomes, which is the number of possible outcomes, calculated using six to the power of the number of dice. With just one hundred dice, the number of outcomes is about the same as the number of atoms in the universe: six-point-five times ten to the seventy-seventh power. Long before you start dealing with big datasets, it becomes computationally impossible to calculate the exact sampling distribution. That means we need to rely on approximations.



**6. Simulating the mean of four dice rolls**

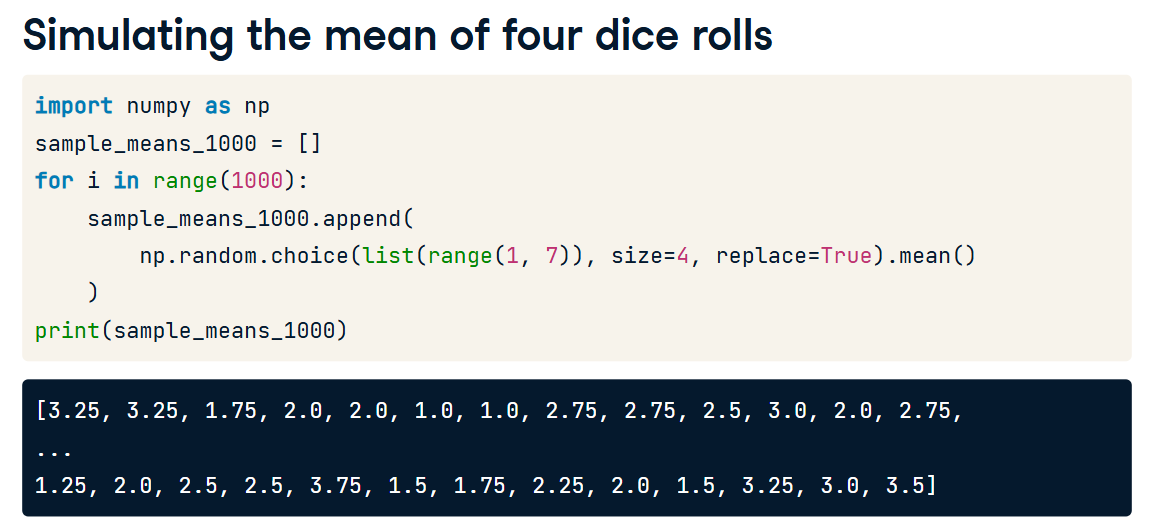
02:19 - 02:47

We can generate a sample mean of four dice rolls using NumPy's random-dot-choice method, specifying size as four. This will randomly choose values from a specified list, in this case, four values from the numbers one to six, which is created using a range from one to seven wrapped in the list function. Notice that we set replace equals True because we can roll the same number several times.

**7. Simulating the mean of four dice rolls**

02:47 - 03:06

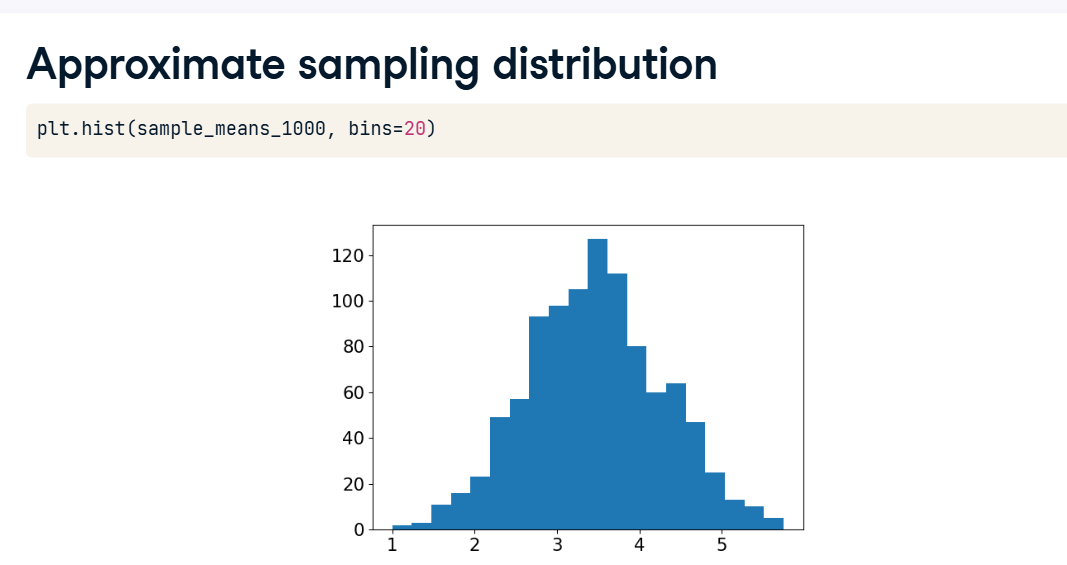
Then we use a for loop to generate lots of sample means, in this case, one thousand. We again use the dot-append method to populate the sample means list with our simulated sample means. The output contains a sampling of many of the same values we saw with the exact sampling distribution.



**8. Approximate sampling distribution**

03:06 - 03:41

Here's a histogram of the approximate sampling distribution of mean rolls. This time, it uses the simulated rather than the exact values. It's known as an approximate sampling distribution. Notice that although it isn't perfect, it's pretty close to the exact sampling distribution. Usually, we don't have access to the whole population, so we can't calculate the exact sampling distribution. However, we can feel relatively confident that using an approximation will provide a good guess as to how the sampling distribution will behave.



**1. Standard errors and the Central Limit Theorem**

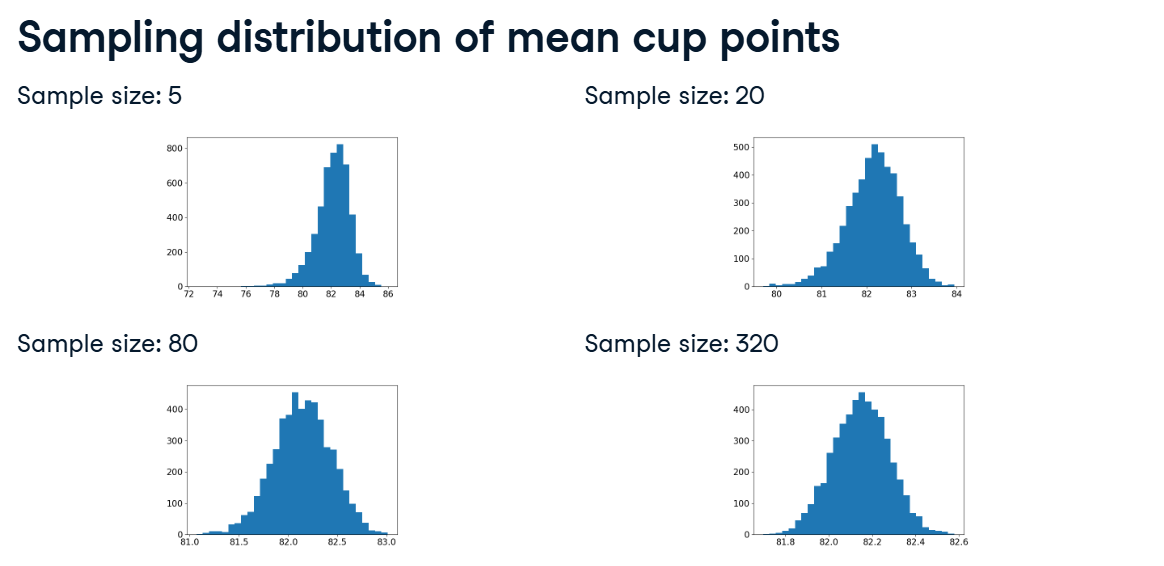
00:00 - 00:14

The Gaussian distribution (also known as the normal distribution) plays an important role in statistics. Its distinctive bell-shaped curve has been cropping up throughout this course.

**2. Sampling distribution of mean cup points**

00:14 - 01:19

Here are approximate sampling distributions of the mean cup points from the coffee dataset. Each histogram shows five thousand replicates, with different sample sizes in each case. Look at the x-axis labels. We already saw how increasing the sample size results in greater accuracy in our estimates of the population parameter, so the width of the distribution shrinks as the sample size increases. When the sample size is five, the x-axis ranges from seventy-six to eighty-six, whereas, for a sample size of three hundred and twenty, the range is from eighty-one-point-six to eighty-two-point-six. Now, look at the shape of each distribution. As the sample size increases, we can see that the shape of the curve gets closer and closer to being a normal distribution. At sample size five, the curve is only a very loose approximation since it isn't very symmetric. By sample size eighty, it is a very reasonable approximation.



**3. Consequences of the central limit theorem**

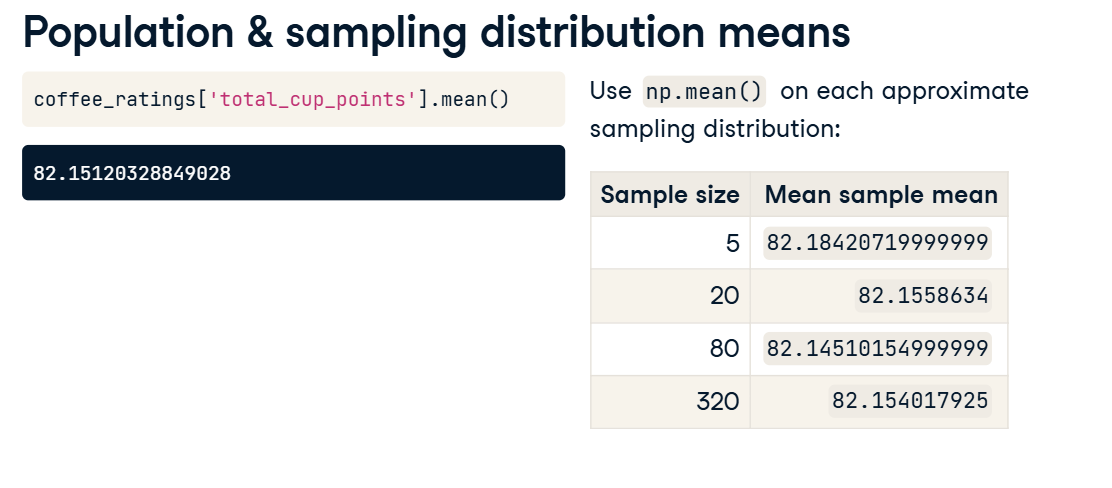
01:19 - 01:42

What we just saw is, in essence, what the central limit theorem tells us. The means of independent samples have normal distributions. Then, as the sample size increases, we see two things. The distribution of these averages gets closer to being normal, and the width of this sampling distribution gets narrower.

**4. Population & sampling distribution means**

01:42 - 02:10

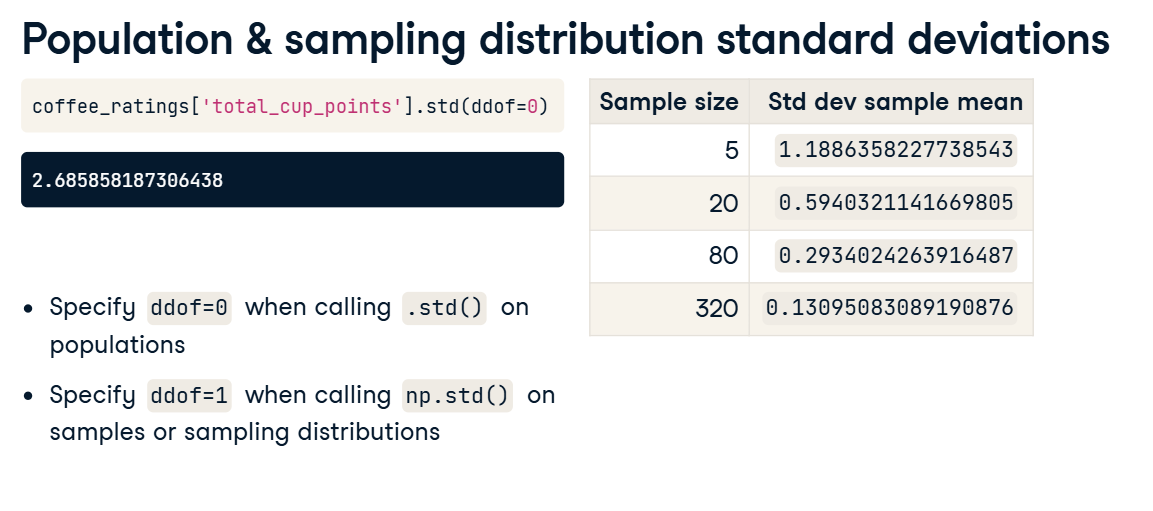
Recall the population parameter of the mean cup points. We've seen this calculation before, and its value is eighty-two-point-one-five. We can also calculate summary statistics on our sampling distributions to see how they compare. For each of our four sampling distributions, if we take the mean of our sample means, we can see that we get values that are pretty close to the population parameter that the sampling distributions are trying to estimate.



**5. Population & sampling distribution standard deviations**

02:10 - 03:04

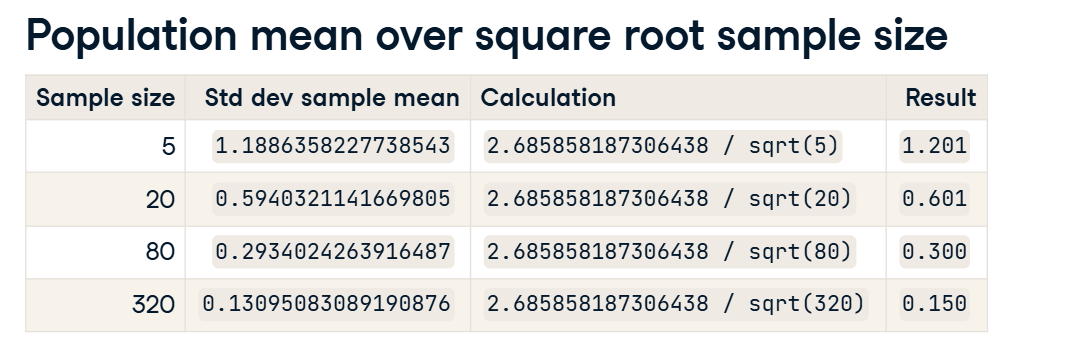
Now let's consider the standard deviation of the population cup points. It's about two-point-seven. By comparison, if we take the standard deviation of the sample means from each of the sampling distributions using NumPy, we get much smaller numbers, and they decrease as the sample size increases. Note that when we are calculating a population standard deviation with pandas dot-std, we must specify ddof equals zero, as dot-std calculates a sample standard deviation by default. When we are calculating a standard deviation on a sample of the population using NumPy's std function, like in these calculations on the sampling distribution, we must specify a ddof of one. So what are these smaller standard deviation values?



**6. Population mean over square root sample size**

03:04 - 03:29

One other consequence of the central limit theorem is that if we divide the population standard deviation, in this case around 2-point-7, by the square root of the sample size, we get an estimate of the standard deviation of the sampling distribution for that sample size. It isn't exact because of the randomness involved in the sampling process, but it's pretty close.



**7. Standard error**

03:29 - 03:52

We just saw the impact of the sample size on the standard deviation of the sampling distribution. This standard deviation of the sampling distribution has a special name: the standard error. It is useful in a variety of contexts, from estimating population standard deviation to setting expectations on what level of variability we would expect from the sampling process.