PROFESSOR: All right. Today I want to spend a few more minutes on plotting, and then return to a subject that will occupy us for a couple of weeks, which is the use of randomness in solving problems. Don't save. All right, so let's first look at one more example of plotting. It's simple. It's so simple you'll find it's not in your handout. So here it is to start with.

PROFESSOR: All I'm doing here is I wrote a little program to show the effect of compound interest, nothing very sophisticated. We start with some principal and an interest rate, and then we just apply it over and over again. And then we're going to plot to show what the principal has become if we just keep compounding the interest. So it is kind of what you'd expect. Compound interest is a nice formula. You can actually get rich applying it, and we see this nice little graph.

On the other hand, we can't really tell what it is. And this is the sort of thing that I see all too often, including my graduate students produce it. They come to my office, they show me a graph, and they start explaining it. And I usually refuse to look at it if it looks like this. There is no point in ever, and I mean ever, producing a graph that does not have a title and labeled axes. And in particular, you have to label the axes to say what they mean.

Fortunately, it's easy enough to do. And here, I've just done that. So I'm going to run the same code to compute the interest, but I'm going to put a title on the graph. You've seen this before, I just want to remind you how it works. PyLab.title And then I'm going to label the x-axis and the y-axis. And that gives me a much more useful graph. Nothing magical here, it's just a reminder that you really need to do these things.

You'll notice here I've not only told you that this is the years of compounding and that this is the principal but I've measured it in dollars. Maybe I should have been even more explicit and said, well, US dollars, whatever. One of the things I did want to point out is you saw the two of these various icons that will let you do things like zoom in on a graph and save a graph. Here's this icon that I think Professor Grimson mentioned, in fact, I know he did. It's a floppy disk, just in case you've never seen one, I brought a floppy disk to show you. This is one of the older floppy disks. These were invented in 1971 by IBM. They were originally 8 inches in diameter and held all of 80 kilobytes of data.

And as you can see, unlike later floppy disks, they actually flopped. Eventually, Apple and others pioneered a non-floppy floppy disk, that was in the '80s. The interesting thing today is I typically carry around a USB stick with me about that big that holds roughly 400,000 times more data than this floppy. And so it's just quite incredible how things have gone along.

All right. I now want to return to what will be the main theme for, as I said, a couple of weeks which is randomness. And in order to talk about randomness we have to talk about probability. And I know that Professor Grimson started down that path just before spring break, but if you're anything like me your mind kind of deteriorated a little bit over spring break, and your head isn't quite into things. And so, I'm just going to back up a tiny bit and start over to get our heads into it, and then fairly quickly move on to new things.

So let's start by asking a simple question. You can tell my head isn't quite yet back to things because I forgot that I needed to begin by gathering chalk. I've now got it. And we'll come over here and take a look at some examples. All right. So the first question I want to ask is, suppose I take a 6-sided die, a fair one, and I roll it 10 times, what's the probability of not getting a single 1, out of that die? Well, how do we go about answering this? Well, there is a wrong way to do it, which is sort of the obvious way, and many people will start down this path. They'll say, well the probability of rolling a 1 on the-- not rolling a 1 on the first try is 1 over 6. Right? That's true? That's not true.

What's the probability of not rolling a 1 the first time? 5 over 6. All right. What's the probability of not rolling a 1 on the second try? 5 over 6. Well, the wrong thing to do, of course, would be to start adding them up. We say, well, OK, we'll just add these up. Well, one way we can tell that's wrong is if we add up 10 of these, we get more than 1. Probabilities can never be more than 1 as we'll see. So let's now try and think of the right way to look at this problem. So you can think about it. If we roll these-- a die 10 times, each time I'll get a number. So I might get a 3, and then a 4, and then a 2. How many possible 10-digit numbers are there? On a 6-sided die, if I roll it 10 times? How many?

AUDIENCE: 6 to the 10th?

PROFESSOR: 6 to the 10th. Exactly Just when we look at binary numbers, if I take a 10-digit binary number, and ask how many different numbers can I represent in 10 binary digits, it's going to be 2 to the 10th. Here we're base 6. So it's going to be 6 to the 10th. Pretty big number. Now I can say, how many of those numbers don't contain a 1? All right. So that's really the question I'm now asking. How many of these don't contain a 1? So as we said, if I look at the first roll the odds of not getting a one the first time is 5 over 6 Now what's the odds of not getting 1 the first or the second time? It's 5 over 6 times 5 over 6.

That makes sense? Because these are independent events. And that's a key notion here. I'm assuming that whether I get a 1 on the second roll is independent of whether I got a 1 on the first roll. It should be true, assuming my dice-- die is fair. Similarly, I can do this for the third roll et cetera. So the probability of not getting a 1 in 10 rolls is going to be (5 over 6) to the 10th. That makes sense? If not, speak up, because things are going to get more complicated quickly.

All right. So that's pretty simple. You-- you all-- are you all with me on that? Now, suppose I ask you the inverse question. What is the probability of getting at least one 1 if I roll the die 10 times? So here I've given you how to compute the probability of not getting any 1's. Suppose I asked you the probability of at least one 1? Yeah?

AUDIENCE: [INAUDIBLE] 1 minus not having a 1?

PROFESSOR: Exactly. Thank you. So that would be 1 minus because we know that the probability, the sum of all the possible things that we can do when we do a probability always has to be 1. It was a good effort. That's it. If you take-- if you want to get something where everything is covered, the probabilities always have to sum to 1. And so now, there are only two possibilities here. One possibility is I don't get any 1's. One possibility is I get at least one 1. So if I take all of the possibilities, and I subtract the possibilities of not getting any 1's, the result must be the probability of getting at least one 1.

This is a very common trick in computing probabilities. Very often when I ask or somebody says, what's the probability of x? The simplest way to compute it, is to compute the probability of not x and subtract it from 1. OK. Again, heading down a wrong track for this, one might have said, well all right, the probability of getting a 1 on the first roll is 1 over 6. The probability of getting a 1 on the second roll is 1 over 6. The probability of getting a third roll is 1 over 6. I'll just add them up, and that will give me the probability of getting at least one one. How do I-- how can I be sure that's wrong? Well when I'm done, I would claim the probability is something like that. And we know that can't be true. Because a probability always has to be less than or equal to 1.

So this is a good trick to keep in mind, whenever you're given a probability problem, try and figure out whether you have a good way to compute it directly, or whether it's simpler to compute the not of the probability, and then subtract it from 1. Probability is really a fun field. It's interesting, it's history, it's intimately connected with the history of gambling. And, in fact, almost all of early probability theory owes its existence to gamblers. People like Cardano, Pascal, Fermat, Bernoulli, de Moivre, Laplace, all famous names you've heard, were motivated by desire to understand games of chance.

Mostly, it started with dice. I've been talking about dice. And in fact, dice are probably the human race's oldest gambling implement. They date at least, archaeologically, to about 600 BC, where a pair of dice was found in Egyptian tombs, actually longer than that. Two millennia before the birth of Christ, people found dice in Egyptian tombs. Typically, they were made from animal bones, but that doesn't matter.

Pascal's interest in it, and Pascal is really considered the founder of probability theory, came when a friend asked him to solve the following problem which I want to work out with you. Is it profitable to bet that given 24 rolls of a pair of fair dice, you would roll a double 6? He actually had a friend who was in the business of gambling, making these bets. So he said, you've got a pair of dice, you roll it 24 times and ask the question, what is the probability of getting what we call today "box cars", in those days they just called two 6's.

This was considered a really hard problem in the mid-17th century. And in fact, Pascal and Fermat, two pretty smart guys as it happens, debated this. They exchanged letters with each other trying to figure out how to solve this problem. It shows how math has advanced because, in fact, today, it's quite an easy problem. So let's work it through and think how would we answer this question. So what's the probability of rolling, of not rolling, a double 6 on the first try?

Well, the probability of not rolling a 6 on one die is a sixth -- 1 over 6. The probability of not rolling a one with the next die is also 1 over 6. So the probability of not getting a die in the first roll, first double 6's is-- the probability of getting a double 6 is 1/36. So the probability of not getting a double 6 is 35/36. Right? So now we know that the probability of not getting it is that. What's the probability of not getting it 24 times in a row? It's that. Which is approximately equal to 0.51. So you can see why the answer was not obvious just by experience. But there is a slight edge in betting that you will not get a double 6 in 24 times. Again, assuming you have fair dice.

As old as dice is, people have built cheater's dice. The excavation of Pompeii, for example, they discovered a pair of loaded dice, dice with a little weight in it so one number would come up more often than it should. And in fact, if you look at the internet today, you will find many sites where you can, let's see, the one I found this morning says quote, "Are you on unusually unlucky when it comes to rolling dice? Investing in a pair of dice that's more reliable might be just what you need." And then it says, "Of course for amusement only." Yeah, we believe that.

All right. As much as I trust probability theory, I don't trust my ability to use it. And so what I did is wrote a little simulation to see if Pascal was right when he did this. So I've got the first-- just this little test roll, which rolls a dice number of times, gets the result. Now, then I decided to check Pascal. So I was going to run 100,000 trials, and keep track of the number of times it worked. So what you'll see I'm doing here is for i in the range number of trials, and this is the way we'll do a lot of these simulations. And in fact, as we deal with probability, we'll be dealing a lot with the notion of simulation, as you are doing in your current problem set.

So for i in range number of trials, for j in range 24, because that was Pascal's friend's game, I'll roll the first die, I'll roll the second die. If they're both 6's I'll say, yes equals 1. And I'll break and then I'll compute the probability of winning or losing. OK? So let's let it rip. So now let's let it rip. There it is. And we can see that it actually comes out pretty close to what Pascal predicted. Should we be surprised that it didn't come out to exactly? Well let's see, is it exactly? What is 35/36 to the 24th?

So that's the-- well, to 17 digits of precision, the exact answer. And you can see we came up with something close to that. Not exactly that, and we wouldn't expect to. Now I only did 100,000 trials. If I did a million trials, I'd probably come up with something even closer, but if I did 2 trials, who knows what I get-- come up with it, right? Could be-- I could get 1, I could get lucky both times, or unlucky. Later on, we'll talk more about the question, how do we know how many trials to run?

Now, the interesting thing is I'm sure it took me less time to write this program than it took Pascal to solve the problem. Now the truth is, I had several hundred years of other people's work to build on. But in general, I think one of the questions you'll find is, is it easier sometimes to write a simulation, than it is to do the probabilities? What I often do in practice is both. I'll scratch my head and figure out how to figure out the answer analytically, and then if it's easy, I'll write some code to simulate the problem, and expect to get roughly the same answer, giving me confidence I've done it correctly.

On the other hand, if I've done the simulation and it had come up with something totally bogus, or totally different, then I would have had to work hard to figure out which was right, the code or the math. Same sort of thing you saw when you looked at the random walk, and the first time it was done an answer showed up that was just wrong. But, you need to have some intuition about a problem, so that you can look at it and say, yeah, that's in the ballpark. And if it's not, it's time to worry.

This kind of simulation that I've just done for the dice game is what's called a "Monte Carlo simulation". It is the most popular kind of simulation named after a Casino on the Riviera, in the small principality of Monaco. This was back in the time when it was hard to find a place you could gamble, and this happened to be one of the places you could. The term was coined in 1949 by Stanislaw Ulam and Nicholas Metropolis, two very well-known mathematicians. Ulam, who later became famous for designing the hydrogen bomb with Teller, invented the method in 1946, and I'm going to quote from his description of how he invented it.

"The first thoughts and attempts I made to practice the Monte Carlo method, were suggested by a question which occurred to me in 1946, as I was convalescing from an illness and playing solitaires. The question was, what are the chances that a canfield solitaire laid out with 52 cards will come out successfully? After spending a lot of time trying to estimate them by pure combinatorial calculations, I wondered whether a more practical method than quote 'abstract thinking' end quote, might not be to lay it out, say, 100 times, and simply observe and count the number of successful plays.

This was already possible to envision with the beginning of the new era of fast computers. And I immediately thought of problems, as you would, I'm sure, immediately thought of problems of neutron diffusion and other questions of mathematical physics. And more generally, how to change processes described by certain differential equations into an equivalent form interpretable as a succession of random operations. Later, I described the idea to John von Neumann, and we began to plan actual calculations."

So as early as 1946, people were thinking about the question of moving away from solving systems of equations, to using randomized techniques to simulate things and try to find out what the actual answer was that way. Now of course "fast" is a relative term. Ulam was probably referring to the ENIAC computer, which could perform about 10 to the 3 additions a second. Not very many, 1,000 operations a second, and weighed approximately 25 tons.

Now today's computers, by comparison, perform 10 to the 9th additions and weigh about 10 to the minus 3 tons. All right. This technique was used during the Manhattan Project to predict what would happen doing-- during nuclear fission and worked. Monte Carlo simulations are an example of what's called "inferential statistics". In brief, and I'm going to be brief because this is not a statistics, course, inferential statistics is based upon one guiding principle. And that principle is that a random sample tends to exhibit the same properties as the population from which it is drawn.

So if I try and sample people, say, for predicting an election, the notion is if I go and I asked a 1,000 people at random in Massachusetts who they're going to vote for, the average will be about the same as if I looked at the whole population. So whenever we use a statistical method like this, so for example, we assumed here, is those 100,000 times I threw the pair of dice, that that would be representative of all possible throws of the dice, the infinite number of possible throws. One always has to ask the question whether this is true, or whether one has a sampling technique that is, for example, giving you a biased sample. Little later in the term, we'll talk about many ways in which you can get fooled here and think you're doing a fair statistical analysis, and get all the math right, and still come up with the wrong answer because this assumption doesn't actually hold.

All right. Let's think about it now in terms of coins, a little simpler than dice, where you can flip a coin and you get either a head or a tail. Suppose Harvey Dent, for example, flipped a coin and it came up heads. Would you feel good inferring from that that the next time he flipped a coin it would also come up heads? I wouldn't. Suppose he flipped it heads and it came up heads twice, in a row. Would you feel comfortable with the third flip would be a head? Probably not.

But suppose he flipped it a 100 times in a row, and it was a head each time. What would you infer? I would infer that the coin two-headed And, in fact, every time it was going to come up heads, because it is so improbable that if it was a fair coin-- what's the probability of having a 100 heads in a row with a fair coin? 1 over what?

AUDIENCE: 1 over 100-- 1 over 2 to the 100th. Right?

PROFESSOR: A half the first time times a half times a half. A huge number, a very small number rather, right? So the probability and a fair coin of getting hundred heads in a row is so low with just 100 flips, that I would begin to think that the coin was not fair. All right. Suppose, however, I flipped it 100 times and I got 52 heads and 48 tails. Well, I wouldn't assume anything from that. Would I assume that the next time I flipped it a 100 times I'd get the same 52 to 48 ratio? Probably not, right? Your common sense tells you you wouldn't. All right. Probably, it tells you, you wouldn't even feel comfortable guessing that there would be more heads than tails the next time.

So when we think about these things, we have to think about the number of tests and how close the answer is to what you would get if you did things at random. This is sort of comparing-- this is technically called comparing something to the null hypothesis. The null hypothesis is what you would get with a random event. And when you do a simulation, if you get something that is far from that, or when you sample a population, you get something that's distant from the null hypothesis, you can assume that maybe you're seeing something real.

All right. Let's look at this in a little less abstract way. So let's go look at some coin flips. So I wrote a simple program, flip. Just flip the coin some number of times and tells me what fraction came up heads. So we can run it, and let's look at a-- suppose I flip a 100, I get 0.55. If I flip 10, I get 0.4. If I flip 10 again, I get 0.5. Now look at that, the same thing twice in a row but now I get 0.2. So obviously, I shouldn't infer too much from 10 flips and even from 100 where I got 0.55. Let's see what happens if I flip 100 again, 0.41, big difference.

So this is suggesting that we can't feel very good about what happens here. Now if I do the following, well I'm feeling a little bit better about this, well for one bad reason and one good reason. The bad reason is, I know the answers 0.5, and these are both close to 0.5, so I feel warm and fuzzy. But that's cheating. I wouldn't need to write the simulation If I knew the answer. But mostly I feel good about it because I'm getting kind of the same answer every time. OK, and that's important. The more I do, the more stable it gets with the larger the number of trials.

This is an example of what's called "the law of large numbers", also known as Bernoulli's Law, after one of the Bernoulli family of mathematicians, and I can't for the life of me remember which Bernoulli. There are a whole bunch of them. Anyway the law states, and it's important to understand this because again it underlies the inferential statistics, that in repeated independent tests, and it's important to note the word "independent", each test has to be independent of the earlier test. In this case, the tests are flips of the coin with the same actual probability we'll call it p, often used to represent probability, of an outcome for each test, the chance that the fraction of times that outcome occurs the outcome that with probability, p, converges to p as number of trials goes to infinity.

All right. So if I did an infinite number of trials, the fraction of heads I would get in this case would be exactly 0.5. Of course I can't do an infinite number of trials. But that's the law of large numbers that says the-- Now, it's worth noting that this law does not imply that if I start out with deviations from the expected behavior, those deviations are likely to be quote "evened out" by opposite deviations in the future. So if I happen to start by getting a whole bunch of heads in a row, it does not mean that I'm more likely to get tails in a subsequent trial. All right. Because if I were-- if that were true, then they wouldn't be independent. Independent means memoryless.

So if I have an independent process, what happens in the future cannot be affected by the past. And therefore, I don't get this business of "they even out". Now people refuse to believe this. If you go to any gambling place, you'll discover that if people threw the roulette wheel, if black comes up 20 times in a row, they'll be a rush to bet on red. Because everyone will say, red is do, red is do, red is do. And every psychologist who has ever done this experiment, finds that people don't believe it. That it's not true. People just don't get probability, and it happens so often it's got a name called "the gambler's fallacy". And there's been great examples of people going broke doing this.

Now notice that the law of large numbers here is about the fraction of times I get an outcome. It does not imply for example, that the absolute difference between the number of heads and the number of tails will get smaller as I run more trials. Right? It doesn't say anything at all about that. It says the ratio of head to tails will approach 1, but not that the difference between them. All right, let's look at an example showing that off.

So what I've got here is this program called "flip plot". This is on your hand out. This is just going to run this business of flipping coins. I should point out just-- I did it this way just to show you. What I'm doing is each flip-- if random.random is less than 5, I'll call it a head, 0.5, I'll call it heads, otherwise a tails. You'll notice that it appears that maybe I'm biasing a little bit, because I'm giving 0.5 a value. But there are so many floating point numbers between 0 and 1, that the probability of getting exactly 0.5 is so small that I can ignore it. It isn't going to really make a difference Random.random is the key issue, the key function that's used to implement all the other random functions that we have in that package.

All right. So I'm going to do it, and I'm going to do it over a range of values. The minimum exponent to the maximum exponent and for exponent in range min x to max x plus 1, I'm going to choose an x value that is 2 to that. So this lets me go over a big range. So I'll see what happens if I get 1 flip, and 2 flips, and 4 and 8 and 16 and 32 et cetera. And then I'm going to just do some plots. I'm going to plot the absolute difference between heads and tails and the ratio of heads to tails.

Let's see what happens when we run that. Actually, probably nothing because I didn't uncomment the run part of it. Let's do that. So I'm going to call flip plot with 4 and 20, running from four trials 2 to the 4 to 2 to the 20. Let's see what we get. Now, you may get different things when you run at different times. In fact, you will. So here we see something kind of uninteresting. Let's cheat and see what we got the first time I ran it, which is on your hand out, and I have a PowerPoint with it. I was-- I knew this might happen. Doesn't usually, but sometimes when you run it you get surprising results.

So here's what happened when I first ran it. Here was the difference between heads and tails. And it seems that, OK, the difference was low, it went up, it went down, it went up, it went down. It seemed to go down dramatically. If you remember what we just saw when I ran it, we also saw something where it went up a little bit then it went down and then shot up dramatically at the end, which was why that scale is so funny. And if we look at the ratio, what we see is it seems to start above 1, drop below 1, and then seems to converge towards 1.

Now, I show this because I want to make a couple of points of plotting. Let's look at this out here. Looks like we have a pretty dramatic trend of this linear drop. Do we? Do we actually have a trend here? Well let's think about it. The default behavior of the plot command in PyLab is to connect points by lines. How many points do I actually have out here? Well, you saw the code. You have the code in the hand out. How many points do you think there are out here? A 1,000? A 100? 3? 2? 2 to 3. Right? Depending on what I mean by "out here".

So what we see here is something that happens a lot. People plot a small number of points, connect them by a line, and mislead the audience into thinking there's a trend when, in fact, maybe all you have is an outlier. So it's problematical here to do it that way. So let's see what happens if we change the code. And what I'm going to do is change it in two ways. Well, maybe I'll change it in one way first. Uncomment. Uncomment.

So what I'm doing here is I am plotting in a different way. This quote "BO" says, don't connect it by lines but just put a dot as an "O" and B says, make it blue. I used blue because it's my favorite color. So now if we look at these things, we'll see something pretty different. So that's the difference between heads and tails, that's the ratio. But now, if we look at the difference between heads and tails here, what we see is it's pretty sparse. So yeah, maybe there's a trend, but maybe not, right? Because way out here I'm only connecting two points, giving an illusion that there is a trend but, in fact, no reason to believe it.

So I always think if you're plotting a small number of points, you're much better off just plotting the points, than you are trying to connect them. Now if we look at this one, again, maybe we'd feel kind of comfortable if there is a trend here, that there are several points on this line. We can't see much of what's going on over here, which gets me to the next thing I want to do is I'm going to use logarithmic axes here. So **PyLab.semilogx** says make the x-axis logarithmic. PyLab.semilogy the y-axis. And so in the case of the absolute difference, I'm going to make both logarithmic. Why am I doing that? Because both have a large range, and by making it logarithmic, I can see what's happening at the left side in this case where things are changing.

When I look at the ratios, the y-axis does not have a very large range, and so there's no need to make it logarithmic. We'll run it. So here, we can see the difference between heads and tails. And now we can see what's going on at the left as we can in Figure (4). And we can see things much more clearly. So log scales can be enormously useful. And in fact, I use them a lot, everyone uses them a lot but, again, it's very important to observe the fact that it's logarithmic and not get fooled. All right. So I talked about linear scaling, logarithmic scaling and we now have charts where I can, perhaps, actually reach some conclusion about what's going on.

The next question is, how certain can I be? Can I really be certain that, indeed, this should be converging to 1? Here, if I sort of look at it, it does look like there's kind of a linear trend of the absolute difference growing as the number of trials grows. How certain can I be of that? You can never get absolute certainty from sampling, because you could never be sure if you haven't been vastly lucky or unlucky. That's not to say you can't get the absolute correct answer. Maybe I could get 0.5, which is the correct answer. But I can't know that that's the correct answer.

So now the question I want to pursue, and it's what we'll cover on Thursday, is what techniques can I use to make a statement of the form, I'm certain within the following range that I have the right answer. That I know the right answer is highly likely to be this close to the answer my simulation is giving me. And we'll look at how we can make those statements and actually believe them. OK, see you on Thursday.