

OPERATOR: The following content is provided under a Creative Commons license. Your support will help MIT OpenCourseWare continue to offer high quality educational resources for free. To make a donation, or view additional material from hundreds of MIT courses, visit MIT OpenCourseWare at ocw.mit.edu.

PROFESSOR: All right, so today we're returning to simulations. And I'm going to do at first, a little bit more abstractly, and then come back to some details. So they're different ways to classify simulation models. The first is whether it's stochastic or deterministic. And the difference here is in a deterministic simulation, you should get the same result every time you run it. And there's a lot of uses we'll see for deterministic simulations. And then there's stochastic simulations, where the answer will differ from run to run because there's an element of randomness in it. So here if you run it again and again you get the same outcome every time, here you may not. So, for example, the problem set that's due today -- is that a stochastic or deterministic simulation? Somebody? Stochastic, exactly. And that's what we're going to focus on in this class, because one of the interesting questions we'll see about stochastic simulations is, how often do you have to run them before you believe the answer? And that turns out to be a very important issue. You run it once, you get an answer, you can't take it to the bank. Because the next time you run it, you may get a completely different answer. So that will get us a little bit into the whole issue of statistical analysis.

Another interesting dichotomy is static vs dynamic. We'll look at both, but will spend more time on dynamic models. So the issue -- it's not my phone. If it's your mother, you could feel free to take it, otherwise -- OK, no problem. Inevitable. In a dynamic situation, time plays a role. And you look at how things evolve over time. In a static simulation, there is no issue with time. We'll be looking at both, but most of the time we'll be focusing on dynamic ones. So an example of this kind of thing would be a queuing network model. This is one of the most popular and important kinds of dynamic simulations. Where you try and look at how queues, a fancy word for lines, evolve over time. So for example, people who are trying to decide how many lanes should be in a highway, or how far apart the exits should be, or what should the ratio of Fast Lane tolls to manually staffed tolls should be. All use queuing networks to try and answer that question. And we'll look at some examples of these later because they are very important. Particularly for things related to scheduling and planning.

A third dichotomy is discrete vs continuous. Imagine, for example, trying to analyze the flow of traffic along the highway. One way to do it, is to try and have a simulation which models each vehicle. That would be a discrete simulation, because you've got different parts. Alternatively, you might decide to treat traffic as a flow, kind of like water flowing through things, where changes in the flow can be described by differential equations. That would lead to a continuous model. Another example is, a lot of effort has gone into analyzing the way blood flows through the human body. You can try and model it discretely, where you take each red blood cell, each white

blood cell, and look at how they move, or simulate how they move. Or you could treat it continuously and say, well, we're just going to treat blood as a fluid, not made up of discrete components, and write some equations to model how that fluid goes through and then simulate that. In this course, we're going to be focusing mostly on discrete simulations.

Now if we think about the random walk we looked at, indeed it was stochastic, dynamic, and discrete. The random walk was an example of what's called a Monte Carlo simulation. This term was coined there. Anyone know what that is? Anyone want to guess? It's the casino, in Monaco, in Monte Carlo. It was at one time, before there was even a Las Vegas, the most famous casino in the world certainly. Still one of the more opulent ones as you can see. And unlike Las Vegas, it's real opulence, as opposed to faux opulence. And this term Monte Carlo simulation, was coined by Ulam and Metropolis, two mathematicians, back in 1949, in reference to the fact that at Monte Carlos, people bet on roulette wheels, and cards on a table, games of chance, where there was randomness, and things are discrete, in some sense. And they decided, well, this is just like gambling, and so they called them Monte Carlos simulations.

What is it that makes this approach work? And, in some sense, I won't go into a lot of the math, but I would like to get some concepts across. This is an application of what are called inferential statistics. You have some sample size, some number of points, and from that you try to infer something more general. We always depend upon one property when we do this. And that property is that, a randomly chosen sample tends to exhibit the same properties as the population from which it is drawn. So you take a population of anything, red balls and black balls, or students, or steps, and at random draw some sample, and you assume that that sample has properties similar to the entire population. So if I were to go around this room and choose some random sample of you guys and write down your hair color, we would be assuming that the fraction of you with blonde hair in that sample would be the same as the fraction of you with blonde hair in the whole class. That's kind of what this means. And the same would be true of black hair, auburn hair, etc. So consider, for example, flipping a coin. And if I were to flip it some number of times, say 100 times, you might be able to, from the proportion of heads and tails, be able to infer whether or not the coin was fair. That is to say, half the times it would be heads, in half the times it would be that tails, or whether it was unfair, that it was somehow weighted, so that heads would come up more than tails. And you might say if we did this 100 times and looked at the results, then we could make a decision about what would happen in general when we looked at the coin.

So let's look in an example of doing that. So I wrote a little program, it's on your handout, to flip a coin. So this looks like the simulations we looked at before. I've got flip trials, which says that the number of heads and tails is 0 for i in x range. What is x range? So normally you would have written, for i in range zero to num flips. What range does, is it creates a list, in this case from 0 to 99 and goes through the list of one at a time. That's fine, but

supposed num flips were a billion. Well, range would create a list with a billion numbers in it. Which would take a lot of space in the computer. And it's kind of wasted. What x range says is, don't bother creating the list just go through the, in this case, the numbers one at a time. So it's much more efficient than range. It will behave the same way as far as the answers you get, but it doesn't use as much space. And since some of the simulations we'll be doing will have lots of trials, or lots of flips, it's worth the trouble to use x range instead of range. Yeah? Pardon?

STUDENT: Like, why would we ever use range instead of x range?

PROFESSOR: No good reason, when dealing with numbers, unless you wanted to do something different with the list. But, there's no good reason. The right answer for the purposes of today is, no good reason. I typically use x range all the time if I'm thinking about it. It was just something that didn't seem worth introducing earlier in this semester. But good question. Certainly deserving of a piece of candy. All right, so for i in x range, coin is equal some random integer 0 or 1. If coin is equal to 0 then heads, else tails. Well, that's pretty easy. And then all I'm going to do here is go through and flip it a bunch of times, and we'll get some answer, and do some plots.

So let's look at an example. We'll try -- we'll flip 100 coins, we'll do 100 trials and see what we get. Error in multi-line statement. All right, what have I done wrong here? Obviously did something by accident, edited something I did not intend edit. Anyone spot what I did wrong? Pardon? The parentheses. I typed where I didn't intend. Which line? Down at the bottom? Obviously my, here, yes, I deleted that. Thank you. All right, so we have a couple of figures here. Figure one, I'm showing a histogram. The number of trials on the y-axis and the difference between heads and tails, do I have more of one than the other on the x-axis. And so we what we could see out of my 100 trials, somewhere around 22 of them came out the same, close to the same. But way over here we've got some funny ones. 100 and there was a difference of 25. Pretty big difference.

Another way to look at the same data, and I'm doing this just to show that there are different ways of looking at data, is here, what I've plotted is each trial, the percent difference. So out of 100 flips. And this is normalizing it, because if I flip a million coins, I might expect the difference to be pretty big in absolute terms, but maybe very small as a percentage of a million. And so here, we can again see that as these stochastic kinds of things, there's a pretty big difference, right? We've got one where it was over 25 percent, and several where it's zero. So the point here, we can see from this graph, that if I'd done only one trial and just assumed that was the answer as to whether my coin was weighted or not, I could really have fooled myself. So the the main point is that you need to be careful when you're doing these kinds of things. And this green line here is the mean. So it says on average, the difference was seven percent.

Well suppose, maybe, instead of, flipping 100, I were to flip 1,000. Well, doesn't seem to want to notice it. One

more try and then I'll just restart it, which is always the safest thing as we've discussed before. Well, we won't panic. Sometimes this helps. If not, here we go. So let's say we wanted to flip 1,000 coins. So now what do we think? Is the difference going to be bigger or smaller than when we flipped 100? Is the average difference between heads and tails bigger or smaller with 1,000 flips than with 100 flips? Well, the percentage will be smaller, but in absolute terms, it's probably going to be bigger, right? Because I've got more chances to stray. But we'll find out. So here we see that the mean difference is somewhere in the twenties, which was much higher than the mean difference for 100 flips. On the other hand, if we look at the percentage, we see it's much lower. Instead of seven percent, it's around two and a half percent in the main.

There's something else interesting to observe in figure two, relative to when we looked at with 100 flips. What else is pretty interesting about the difference between these two figures, if you can remember the other one? Yeah?

STUDENT: There are no zeros?

PROFESSOR: There are no zeros. That's right, as it happens there were no zeros. Not so surprising that it didn't ever come out exactly 500, 500. What else? What was the biggest difference, percentage-wise, we saw last time? Over 25. So notice how much narrower the range is here. Instead of ranging from 2 to 25 or something like that, it ranges from 0 to 7, or maybe 7 and a little. So, by flipping more coins, the experiment becomes more reproducible. It looks like the same because of scaling, but in fact the range is much narrower. Each experiment tends to give an answer closer to all the other experiments. That's a good thing. It should give you confidence that the answers are getting pretty close to right. That they're not bouncing all over the place. And if I were to flip a million coins, we would find the range would get very tight. So notice that even though there's similar information in the histogram and the plot, different things leap out at you, as you look at it.

All right, we could ask a lot of other interesting questions about coins here. But, we'll come back to this in a minute and look at some other questions. I want to talk again a little bit more generally. It's kind of easy to think about running a simulation to predict the future. So in some sense, we look at this, and this predicts what might happen if I flipped 1,000 coins. That the most likely event would be that I'd have something under 10 in the difference between heads and tails, but that it's not terribly unlikely that I might have close to 70 as a difference. And if I ran more than 100 trials I'd see more, but this helps me predict what might happen.

Now we don't always use simulations to predict what might happen. We sometimes actually use simulations to understand the current state of the world. So for example, if I told you that we are going to flip three coins, and I wanted you to predict the probability that all three would be either heads, or all three would be tails. Well, if you'd studied any probability, you could know how to do that. If you hadn't studied probability, you would say, well, that's OK, we have a simulation right here. Let's just do it. Here we go again. And so let's try it. Let's flip three coins, and

let's do it 4,000 times here. Well, that's kind of hard to read. It's pretty dense. But we can see that the mean here is 50. And, this is a little easier to read. This tells us that, how many times will the difference, right, be zero 3,000 out of 4,000. Is that right? What do you think? Do you believe this? Have I done the right thing? three coins, 4,000 flips, how often should they all be heads, or how often should they all be tails? What does this tell us? It tells us one-fourth of the time they'll all be-- the difference between, wait a minute, how can the difference between -- something's wrong with my code, right? Because I only have two possible values. I hadn't expected this. I obviously messed something up. Pardon?

STUDENT: It's right.

PROFESSOR: It's right, because?

STUDENT: Because you had an odd number of flips, and when you split them --

PROFESSOR: Pardon?

STUDENT: When you split an odd number --

PROFESSOR: Exactly. So it is correct. And it gives us what we want. But now, let's think about a different situation. Anybody got a coin here? Anyone give me three coins? I can trust somebody, I hope. What a cheap -- anybody got silver dollars, would be preferable? All right, look at this. She's very carefully given me three pennies. She had big, big money in that purse, too, but she didn't want me to have it. All right, so I'm going to take these three pennies, jiggle them up, and now ask you, what's the probability that all three of them are heads? Anyone want to tell me? It's either 0 or 1, right? And I can actually look at you and tell you exactly which it is. And you can't see which it is. So, how should you think about what the probability it? Well, you might as well assume that it's whatever this graph tells you it is. Because the fact that you don't have access to the information, means that you really might as well treat the present as if it's the future. That it's unknown. And so in fact we frequently, when there's data out there that we don't have access to, we use simulations and probabilities to estimate, make our best guess, about the current state of the world. And so, in fact, guessing the value of the current state, is really no different from predicting the value of a future state when you don't have the information.

In general, all right, now, just to show that your precautions were unnecessary. Where was I? Right. In general, when we're trying to predict the future, or in this case, guess the present, we have to use information we already have to make our prediction or our best guess. So to do that, we have to always ask the question, is past behavior a good prediction of future behavior? So if I flip a coin 1,000 times and count up the heads and tails, is that a good prediction what will happen the next time? This is a step people often omit, in doing these predictions. See the recent meltdown of the financial system. Where people had lots of stochastic simulations predicting what the

market would do, and they were all wrong, because they were all based upon assuming samples drawn from the past would predict the future. So, as we build these models, that's the question you always have to ask yourself. Is, in some sense, this true? Because usually what we're doing is, we're choosing a random sample from the past and hoping it predicts the future. And that is to say, is the population we have available the same as the one in the future. So it's easy to see how one might use these kinds of simulations to figure out things that are inherently stochastic. So for example, to predict a poker hand. What's the probability of my getting a full house when I draw this card from the deck? To predict the probability of coming up with a particular kind of poker hand. Is a full house more probable than a straight? Or not? Well, you can deal out lots of cards, and count them up, just as Ulam suggested for Solitaire. And that's often what we do.

Interestingly enough though, we can use randomized techniques to get solutions to problems that are not inherently stochastic. And that's what I want to do now. So, consider for example, pi. Many of you have probably heard of this. For thousands of years, literally, people have known that there is a constant, pi, associated with circles such that pi times the radius squared equals the area. And they've known that pi times the diameter is equal to the circumference. So, back in the days of the Egyptian pharaohs, it was known that such a constant existed. In fact, it didn't acquire the name pi until the 18th century. And so they called it other things, but they knew it existed. And for thousands of years, people have speculated on what its value was. Sometime around 1650 BC, the Egyptians said that pi was 3.16, something called the Rhind Papyrus, something they found. Many years later, about 1,000 years later, the Bible said pi was three. And I quote, this is describing the specifications for the Great Temple of Solomon. "He made a molten sea of 10 cubits from brim to brim, round in compass, and 5 cubit the height thereof, and a line of 30 cubits did compass it round about." So, all right, so what we've got here is, we've got everything we need to plug into these equations and solve for pi, and it comes out three. And it does this in more than one place in the Bible. I will not comment on the theological implications of this assertion. Sarah Palin might. And Mike Huckabee certainly would.

The first theoretical calculation of pi was carried out by Archimedes, a great Greek mathematician from Syracuse, that was about somewhere around 250 BC. And he said that pi was somewhere between 223 divided by 71, and 22 divided by 7. This was amazingly profound. He knew he didn't know what the answer was, but he had a way to give an upper and a lower bound, and say it was somewhere between these two values. And in fact if you calculate it, the average of those two values is 3.1418. Not bad for the time. This was not by measurement, he actually had a very interesting way of calculating it. All right, so this is where it stood, for years and years, because of course people forgot the Rhind Papyrus, and they forgot Archimedes, and they believed the Bible, and so three was used for a long time. People sort of knew it wasn't right, but still. Then quite interestingly, Buffon and Laplace, two great French mathematicians, actually people had better estimates using Archimedes' methods long before they came along, proposed a way to do it using a simulation. Now, since Laplace lived between 1749 and 1827, it

was not a computer simulation. So I'm going to show you, basically, the way that he proposed to do it. the basic idea was you take a square, assume that's a square, and you inscribe in it a quarter of a circle. So here, you have the radius of the square r . And then you get some person to, he used needles, but I'm going to use darts, to throw darts at the shape. And some number of the darts will land in the circle part, and some number of the darts will land out here, in the part of the square that's not inscribed by the circle. And then we can look at the ratio of the darts in the shaded area divided by the total number of darts in the square. And that's equal to the shaded area divided by the area of the square. The notion being, if they're landing at random in these places, the proportion here and not here will depend upon the relative areas. And that certainly makes sense. If this were half the area of the square, then you'd expect half the darts to land in here.

And then as you can see in your handout, a little simple algebra can take this, πr^2 equals the area, you can solve for π , and you can get that π is equal to $4h$, and I'll write it, h , where h is the hit ratio, the number falling in here. So people sort of see why that should work intuitively? And that it's a very clever idea to use randomness to find a value that there's nothing random about. So we can now do the experiment. I need volunteers to throw darts. Come on. Come on up. I need more volunteers. I have a lot of darts. Anybody else? Anybody? All right, then since you're all in the front row, you get stuck. So now we'll try it. And you guys, we'll see how many of you hit in the circle, and how many of you hit there. Go ahead, on the count of 3, everybody throw. 1, 2, 3. Ohh! He did that on purpose. You'll notice Professor Grimson isn't here today, and that's because I told him he was going to have to hold the dart board. Well, what we see here is, we ignore the ones that missed all together. And we'll see that, truly, I'm assuming these are random throws. We have one here and two here. Well, your eyes will tell you that's the wrong ratio. Which suggests that having students throw darts is not the best way to solve this problem. And so you will see in your handout a computer simulation of it.

So let's look at that. So this is find pi. So at the beginning of this code, by the way, it's not on your handout, is some magic. I got tired of looking at big numbers without commas separating the thousands places. You've see me in other lectures counting the number of zeros. What we have here is, that just tells it I have to have two versions, one for the Mac, and one for the PC. To set some variables that had to write integers, things in general, and I'm saying here, do it the way you would do it in the United States in English. And UTF8 is just an extended character code. Anyway, you don't need to learn anything about this magic, but it's just a handy little way to make the numbers easier to read. All right, so let's let's try and look at it. There's not much interesting to see here. I've done this little thing, `format ints`, that uses this magic to say grouping equals true, that means put a comma in the thousand places. But again, you can ignore all that. The interesting part, is that from Pylab I import star, import random in math. As some of you observed, the order these imports matters. I think I sent out an email yesterday explaining what was going on here. This was one of the things that I knew, and probably should've mentioned. But since I knew it, I thought everyone knew. Silly me. It was of course a dumb thing to think. And then I'm going to

throw a bunch of darts. The other thing you'll notice is, throw darts has a parameter called should plot. And that's because when I throw a billion darts, I really don't want to try and take the time to plot a billion points. So let's first look at a little example. We'll try throwing 10,000 darts. And it gives me an estimated value of pi of 3.16. And what we'll see here, is that the number of darts thrown, the estimate changes, right? When I threw one dart, the estimate of pi was 4. I threw my second dart, it dropped all the way to 3. And then it bounced around a while, and then at the end, it starts to really stabilize around the true value. You'll notice, by the way, that what I've got here is a logarithmic x-axis. If you look at the code, you'll see I've told it to be semi log x. And that's because I wanted you to be able to see what was happening early on, where it was fluctuating. But out here it's kind of boring, because the fluctuations are so small. So that was a good way to do it. All right now. Do I think I have enough samples here? Well, I don't want you to cheat and look at the estimate and say no, you don't, because you know that's not the right answer. And, it's not even as good as Archimedes did. But how could you sort of look at the data, and get a sense that maybe this is not the right answer? Well, even at the end, if we look at it, it's still wiggling around a fair amount. We can zoom in. And it's bouncing up and down here. I'm in a region, but it's sort of makes us think that maybe it hasn't stabilized, right? You'd like it to not be moving very much. Now, by the way, the other thing we could've looked at, when we ran it, let's run it again, probably get a different answer by the way. Yeah, notice the different answer here. Turns out to be a better answer, but it's just an accident, right?

Notice in the beginning it fluctuates wildly, and it fluctuates less wildly at the end. Why is that? And don't just say because it's close to right and it knows it. Why do the mathematics of this, in some sense, tell us it has to fluctuate less wildly at the end? Yes?

STUDENT: [INAUDIBLE]

PROFESSOR: Exactly, exactly right. If I've only thrown two darts, the third dart can have a big difference in the average value. But if I've thrown a million darts, the million and first can't matter very much. And what this tells me is, as I want ever more digits of precision, I have to run a lot more trials to get there. And that's often true, that simulations can get you in the neighborhood quickly, but the more precision you want, the number of steps grows quite quickly. Now, the fact that I got such different answers the two times I ran this suggests strongly that I shouldn't believe either answer. Right? So we need to do something else. So let's try something else. Let's try throwing a lot more darts here, and see what we get. Now if you look at my code, you'll see I'm printing intermediate values. Every million darts, I'm printing the value. And I did that because the first time I ran this on a big number, I was afraid I had an infinite loop and my program was not working. So I just said, all right, let's put a print statement in the loop, so I could see that it's making progress. And then I decided it was just kind of nice to look at it, to see what was going on here. So now you see that if I throw 10 million darts, I'm starting to get a much better estimate. You'll also see, as predicted, that as I get further out, the value of the estimate changes less and

less with each million new darts, because it's a smaller fraction of the total darts. But it's getting a lot better. Still not quite there. Let's just see what happens, I can throw in another one. This is going to take a little while. So I can talk while it's running. Oops, what did I do here?

So, it's going to keep on going and going and going. And then if we were to run it with this number of darts several times over, we would discover that we got answers that were very, very similar. From that we can take comfort, statistically, that we're really getting close to the same answer every time, so we've probably thrown enough darts to feel comfortable that we're doing what's statistically the right thing. And that there maybe isn't a lot of point in throwing more darts. Does that mean that we have the right answer? No, not necessarily, and that's what we're going to look at next week.