Episode Book

EPISODE #6206

Monte Carlo: How Random Guesses Solve Impossible Problems

PUBLISHED	22 October 2026 at 21:32 UTC
DURATION	22:39
LISTEN	myweirdprompts.com/episode/monte-carlo-random-sampling-explained-58216/
HOSTS	Corn, Herman, Dorothy, Hilbert
SCRIPT MODEL	anthropic/claude-sonnet-5.5
VOICES	chatterbox-regular (chatterbox-local)
PIPELINE	V5.19
BOOK GENERATED	23 October 2026 at 07:44 UTC

SYNOPSIS

Monte Carlo methods turn chance into answers, and they were born from a card game at Los Alamos in 1946. This episode traces the method from Stanislaw Ulam's solitaire problem through John von Neumann's neutron diffusion work, the ENIAC calculation, and Buffon's needle — then digs into the practical question everyone hits: how many runs do you actually need? The square-root rule, pilot runs, sequential stopping, and why rare events blow up your compute budget all get unpacked in plain terms.

TRANSCRIPT



Corn

...and the part that bothers me is that it sounds like cheating. You don't solve the hard problem. You just try it a few thousand times and see what happens.



Herman

It is cheating, in the most respectable way possible. And it's what Daniel wants taken apart from the ground up: where Monte Carlo came from, why random sampling gives you a real answer at all, and how anyone decides how many repeat runs are enough. In terms that don't need a mathematics degree.



Corn

Good, because my highest qualification is in napping. Where does it start?



Herman

With a card game. Nineteen forty-six, Stanislaw Ulam, a mathematician at Los Alamos, is recovering from an illness and playing solitaire. Canfield solitaire, the one with fifty-two cards where you win rarely. And he wonders what the odds are that a layout comes out. He tries to work it out the proper way, counting arrangements, and it's hopeless, there are too many.



Corn

So he cheats.



Herman

He asks a different question. What if I just deal it a hundred times and count how many come out? He realises he can answer an analytic question by running the experiment. Then the thought jumps. If that works for cards, it works for any process with chance in it. Like a neutron wandering through a lump of material.



Corn

Which is what Los Alamos cared about.



Herman

It's the whole job. A neutron flies some distance, hits a nucleus, and then maybe it scatters, maybe it's absorbed, maybe it triggers a fission and makes more neutrons. Each step is a dice roll with known odds. Nobody can write one clean equation for the whole chain, but you can follow one imaginary neutron through, roll the dice at each step, write down what happened, then do it for the next neutron. Do that enough times and the average of the imaginary neutrons tells you about the real thing.



Corn

And Ulam told John von Neumann.



Herman

Later in that same year, yes. Von Neumann got it immediately, and within a few months he was working out how you'd do neutron diffusion this way. The name comes from Nicholas Metropolis, who suggested Monte Carlo. As Metropolis later told it, the casino reference was because Ulam had an uncle who kept borrowing money from relatives to gamble. I should say that's the story as recollected decades afterwards, so I wouldn't swear to it.



Corn

Of all the names for a method that involves a sloth with a pencil, a casino is the glamorous one.



Herman

Fermi had actually done something like it in the nineteen thirties, working out neutron diffusion by sampling, and apparently never published it. And later he proposed a little mechanical trolley called the FERMIAC. It rolled across a scale drawing of a reactor and traced a neutron's path, with the settings dialled in from random numbers. A physical Monte Carlo machine.



Corn

A toy car that does the dice rolls for you.



Herman

It's in a museum at Los Alamos. The real machine was ENIAC, though. In spring nineteen forty-eight, von Neumann, Metropolis and others ran what's usually called the first fully automated Monte Carlo calculation on it, on the behaviour of a fission weapon core. Klara von Neumann, John's wife, is credited with a large part of the coding. Sources disagree on the exact month, so spring is as tight as I'll go.



Corn

And then they wrote it up.



Herman

Metropolis and Ulam, nineteen forty-nine, "The Monte Carlo Method", in a physics journal, I think, which is where the name got out into the world. Though the idea had cousins. Buffon, in seventeen seventy-seven, asked what happens if you drop a needle onto a floor of parallel boards. The chance it lands across a crack depends on pi. So you drop a few thousand needles, count the crossings, and you get a rough value of pi out of a bag of sewing needles.



Corn

That's a terrible way to get pi.



Herman

Appallingly slow. But look at what he did. A deterministic number, pi, estimated by chance. And there's a famous bad apple in the story. In nineteen oh one, an Italian named Lazzarini reported three thousand four hundred and eight tosses that produced three hundred fifty-five over one hundred thirteen, which is pi correct to six decimal places.



Corn

With three thousand needles? That's lucky.



Herman

It's suspiciously lucky, and nobody believes it was run honestly. The number of tosses fits the answer a bit too neatly. Whether he stopped when the result looked good or worked backwards, I can't say. It's a good cautionary tale, and we'll come back to it when we get to deciding when to stop.



Corn

Right, so the cards, the neutrons, the needles all share a trick. Why does it work? Why should dealing a hundred hands tell me about the real odds?



Herman

Because of the law of large numbers, which you already believe. Flip a fair coin ten times and you might get seven heads. Flip it ten thousand times and you'll be very close to half. The more runs, the closer the average gets to the truth. The part that matters for planning is how fast it gets there.



Corn

Slowly, I'm guessing, since I'm about to be asked to do arithmetic.



Herman

Slowly, and in a very particular way. The error shrinks with the square root of the number of runs. Four times as many runs halves your error. A hundred times as many runs buys you one extra decimal place.



Corn

So if I want to be twice as sure I pay four times as much.



Herman

Which is the thing every Monte Carlo user eventually curses. Precision is expensive and the price goes up fast. Want ten times the accuracy? A hundred times the runs. Want a hundred times the accuracy? Ten thousand times the runs.



Corn

It's like buying ever-thinner slices of the same cake. You pay full price and get less each time.



Herman

Fair. Now here's the surprise, and it's why the method won. That square root rule doesn't care how complicated your problem is. Say you want the volume of a strange shape. The old way is a grid: chop the space into little cells, measure each one. In two dimensions, ten points along each side gives you a hundred cells. Fine. But a problem with ten variables, and ten points per variable, is ten billion cells.



Corn

And a problem with a hundred variables?



Herman

More cells than anyone could count. Grids blow up as you add dimensions. Monte Carlo doesn't. Throw random darts into the ten-variable space or the hundred-variable space and your error shrinks by the same square root of the number of darts. How wild the shape is affects the constant out front, but not the rule.



Corn

So a pile of random darts beats a careful grid.



Herman

Say it were your problem, Corn, a hundred variables, would you trust the grid or the darts?



Corn

Darts. A grid with that many cells would outlive me, and I'm a sloth.



Herman

In high dimensions, by a mile. In one or two dimensions, a good grid wins easily, so I'm not saying never use one. But a weapon core has a lot of variables, and so does a financial portfolio, and so does an AI evaluation. That's the reason it spread from physics into everything.



Corn

Which lets me ask the practical question. How many runs? Because that's the one that got us in trouble with the geopolitical simulation, the episode where a hundred and fifty dollars turned into fifteen thousand when it scaled. They ran a hundred runs and called it a day.



Herman

Okay, let's check a hundred. Take the simplest case: you're estimating a probability, say the chance the simulation ends in some outcome, and you're totally in the dark, so call it fifty-fifty. The uncertainty on your estimate is roughly the spread of a single run divided by the square root of the number of runs. For a yes-or-no outcome at fifty percent, a single run has a spread of a half. Divide by ten, because the square root of a hundred is ten, and you get five points. Then you double it to cover ninety-five percent confidence, and the margin is about plus or minus ten percentage points.



Corn

So a hundred runs says it came out yes fifty percent of the time, give or take ten.



Herman

Anywhere from forty to sixty. That's the whole story from a hundred runs. If the question was whether a policy beats another one by three points, a hundred runs cannot tell you.



Corn

That's quite a lot of money for forty to sixty.



Herman

Now flip the question and ask what margin you want. Plus or minus five points needs three hundred eighty-five runs. Plus or minus three points needs one thousand sixty-eight. Plus or minus one point, nine thousand six hundred and four.



Corn

Hold on, that's a big jump. Five points to one point is twenty-five times the runs?



Herman

Twenty-five times, because the rule is a square. Five times tighter costs five squared. You got it faster than I did the first time I did it.



Corn

Herman. Is there any way to know how many runs before I've done the runs? It sounds circular.



Herman

It is slightly circular, and the fix is a pilot. Run a small batch first, say fifty or a hundred, just to see how spread out the results are. That spread is the number you couldn't know in advance. Then you plug it into the formula: the number of runs you need is the confidence multiplier times the spread, divided by the margin you can tolerate, all squared. The pilot isn't the answer, it's the measuring stick.



Corn

Which assumes I only decide once.



Herman

You don't have to. The other approach is to run in batches and check the interval after each one. Keep going until the margin is tight enough, then stop. It's called sequential stopping, and it's the sensible approach when nobody can predict the spread.



Corn

That sounds like exactly what Lazzarini did. Keep dropping needles until it looks right and then stop.



Herman

That is the catch, and good that you saw it. If you peek after every batch and stop the moment you like what you see, you overstate your confidence. You gave luck many chances to look good and you quit while it did. The remedy is modest: set the margin in advance, require a bit more than you think you need, and don't stop on a single lucky glance.



Corn

So Lazzarini is a case of peeking.



Herman

Possibly peeking, possibly worse. I'd rather not accuse a man dead a hundred years on a technicality. The numbers are a bit too tidy, that's all I'll say.



Corn

Does the same arithmetic cover the other question, the one from that episode about how many runs an AI eval needs, where the noise between runs mattered more than anything?



Herman

Same engine. Run-to-run noise is the spread in the formula. If one run of a model gives you a different answer each time, that's your sigma, and the number of runs you need follows from it. The difference is that evals have a second source of variation, which prompts you chose, and no amount of repeated runs fixes that. Runs only shrink the first kind.



Corn

And the rare stuff? Because the interesting events in a simulation are rarely fifty-fifty.



Herman

Rare events are where naive run counts fall apart. Say something happens one time in ten thousand. For the estimate to be any good, you need to see it happen many times, because an estimate built on two sightings is basically a guess. The working rule is you need on the order of a thousand or more hits, not a thousand runs. And to see a thousand hits of a one in ten thousand event, you need ten million runs.



Corn

What about one in a hundred million?



Herman

To get within about five percent of the true value, you're looking at roughly a hundred and fifty billion runs. And that's before you've paid for a single one of them.



Corn

Hm. That's a lot of neutrons.



Herman

It's why people who care about catastrophic failures don't use plain Monte Carlo. We'll get there. But there's a related trick for the other extreme. If you ran a thousand trials and saw zero failures, you don't get to say the failure rate is zero. The rule of three says the real



Herman (continued)

rate could still be as high as three divided by the number of trials, at ninety-five percent confidence. A thousand clean trials only rules out a failure rate above about three in a thousand.



Corn

So zero out of a thousand is "less than three in a thousand", not "never". That's a surprisingly mean result.



Herman

It's an honest one. Have you ever had a situation where you'd tested something a lot and it never failed, and you felt more sure than the testing allowed?



Corn

Constantly. A thousand naps, zero failures. Apparently I can only be ninety-five percent sure the schedule is fine.



Herman

Well, that is all from us on Monte Carlo methods, but before we go, it looks like we have another call on the line.

D

Dorothy

Hello dears, it is Dorothy, and this is Dorothy's Kitchen.



Corn

Mum.

D

Dorothy

Today I am going to teach you how to make a cup of tea. Bag left in.



Herman

I'm so glad you rang, Dorothy.

D

Dorothy

The ingredients are one tea bag. Water. Milk, if you must. And the tea bag stays in the cup, that is the whole secret, sweetheart.



Corn

Does the water go in first, Mum?

D

Dorothy

Now the method, and pay attention, because timing is everything.



Herman

I'm writing this down, Dorothy.

D

Dorothy

Boil the water. Pour it in. Then walk away. Do not lift the bag out. Do not squeeze it. Things have gone wrong before when people squeezed it, and I would rather not say how.



Herman

How long does it sit?

D

Dorothy

Let it sit until it is the colour of a good strong brick, then add the milk and drink it before it goes cold. Now be a good boy and eat something proper, Corn. Goodbye, dear.



Corn

Bye, Mum.



Herman

Right, correlated runs. This one catches a lot of people, and it's where the formula I gave you earlier quietly stops being true.



Corn

Stops being true how?



Herman

Everything so far assumed each run is independent, like a fresh coin flip. A lot of real simulations aren't. Imagine a method where each sample is a small nudge from the previous one, a wandering chain. That's how the Markov chain version works, a descendant of nineteen fifty-three work by Metropolis and colleagues. Consecutive samples are cousins, not strangers. Ten thousand of them tell you less than ten thousand independent ones, sometimes far less.



Corn

So you've paid for ten thousand runs and received, what, a few hundred worth of information?



Herman

Sometimes, yes. The fix is the effective sample size. Take the number of runs and divide by one plus twice the sum of the correlations between runs at different gaps. The more strongly each run resembles the last, the smaller the effective number. Then you use that number in the error formula, not the raw count.



Corn

And on the AI side, that's when you run a model with the same seed or a similar prompt a hundred times and treat them as a hundred separate opinions.



Herman

It's the same mistake, yes. A hundred correlated runs aren't a hundred runs. In my view that's the number one way people overstate how reliable a simulation is. They count the rows in the spreadsheet.



Corn

Okay. So runs are expensive, they scale with a square root, and the rare events are practically impossible. Is there a cheaper route?



Herman

Yes, and it's the part of the field that's seen real cleverness. The goal is the same reliability from fewer runs, which people call variance reduction. There are four I'd name.



Corn

Go on, I'll hold the clipboard.



Herman

The first is antithetic variates. Whenever you draw a random number, you also use its mirror image. If you drew point three, you also draw point seven. If your first run came out too high, the mirrored one tends to come out too low, and when you average the pair the errors cancel. It's like weighing yourself on two bathroom scales, one that reads heavy and one that reads light.



Corn

Which only works if the problem moves in one direction.



Herman

For problems that rise or fall steadily with the input, yes. Second is stratification. Rather than letting chance decide where your samples land, you divide the range into slices and make sure each slice gets its share. If you're polling a country, you don't want chance to hand you everyone from one town. You make sure every town is represented.



Corn

A fair-minded sloth would have thought of that.



Herman

Third is importance sampling, which is the answer to the hundred-and-fifty-billion-runs problem. Rather than waiting around for the rare event, you deliberately sample the region where it happens, far more often than it really does, and then correct the count by the known odds. Imagine surveying a village that's tiny but matters a great deal, so you interview a lot more of its residents than their population would suggest, then weight their answers back down.



Corn

And the correction is the clever part, because otherwise you've just told yourself a lie.



Herman

The correction is the entire method. Get the weights right and you can see a one in a hundred million event in thousands of runs rather than billions. Get the weights wrong, and you've built a very expensive random number generator.



Corn

And the fourth?



Herman

Quasi-Monte Carlo. This one abandons randomness. Instead of random points you use points that are deliberately spread out evenly, with names like Sobol and Halton sequences. Picture seating guests at a wedding so no table is crowded and none is empty, compared with letting everyone pick their own chair. For smooth problems with a moderate number of variables, the error can fall roughly like one over the number of runs instead of one over the square root. Four times the runs doesn't halve it, it quarters it.



Corn

Which is a lot of improvement for taking away the dice.



Herman

Would you take the evenly spread points or the dice, Corn, if it were your problem?



Corn

Spread points, as long as nobody makes me carry them. Dice feel like a lot of effort for the same answer.



Herman

It's a strange sort of victory for the dice, honestly. After seventy-odd years, one of the best upgrades is to stop being random. And it tells you what randomness was always doing: it was a cheap way to cover the space. The covering is the point. And the next problem is the dice themselves. The numbers are never truly random, they come out of a formula, a pseudo-random generator, and if you wanted genuinely random digits you'd have to go and roll a physical die, over and over, and...



Hilbert

Pay Desmond Fairweather. Twenty-two pence a digit, by postcard, from the flat above a launderette in Walthamstow. Herman, you've done the square root three ways already, so I'll be quick. I wanted proper random digits in two thousand three and Desmond said he had them.



Corn

One digit per postcard.



Hilbert

One a day. Desmond doesn't roll them. His wife Maureen rolls them at the kitchen table, with a die that has two sixes on it. Desmond signs the cards, and the signing is extra. Hold on, Corn, your microphone's going on the hard consonants, back off an inch. Anyway, the die. She says she knows which ones are the sixes.



Herman

A die with two sixes is just a biased sampler. Reweight the sixes by half and you have a fair die again, which is importance sampling in miniature.



Hilbert

Statistics journal, that nineteen forty-nine paper, by the way. The American Statistical Association. I've got a copy in the hall, from a different supplier. I'm at four hundred and six cards. A thousand digits takes him nearly three years. They're in a biscuit tin by the kettle, and every morning the top card decides which bus I get. Today's was a three.



Herman

Journal of the American Statistical Association, yes, I had that backwards. Which makes sense in hindsight, since the people who most needed the method were statisticians.



Corn

Three years for a thousand digits is a pilot study, not a sample.



Herman

A thousand digits is a perfectly decent sample for plenty of questions. It's just slow.



Corn

Slow and honest. Most simulations pick fast and sloppy.



Herman

A loose end I liked: William Gosset, the man who wrote as Student, used a card experiment in nineteen oh eight. Three thousand slips of cardboard with measurements on them, shuffled and drawn at random, to see how sample averages behave. I've only seen that in secondary sources, so take it as how I understand it. But it means the dealing-cards-to-learn-about-chance idea predates Ulam's solitaire by nearly forty years.



Corn

Ulam just happened to be bored in the right building.



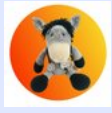
Herman

At the right moment, with a computer coming. Cheap, fast runs have made it easy to forget the arithmetic, so what I'd keep is asking what a run costs, in money and in days, before ordering another thousand. A digit a day concentrates the mind.



Corn

Thanks to Hilbert Flumingtop for producing, and to Daniel for the prompt, which made me multiply in public.



Herman

This has been My Weird Prompts. If you want more, you'll find the whole archive at my weird prompts dot com.



Corn

See you tomorrow.