

RUMOURS AND ERRORS

Brian Hayes

The confessional essay is not a popular genre in mathematics and the sciences; few of us wish to dwell on our mistakes or call attention to them. An inspiring exception is Donald E. Knuth of Stanford University. During a decade's labor on the T_EX typesetting system, he kept a meticulous log of all his errors, and then he published the list with a detailed commentary.

I have long admired Knuth's act of public bravery, and this column is my attempt to follow his example. I took courage from the thought that if there is any realm of life in which I might hope to surpass Don Knuth, it's in making mistakes; but, alas, I've fallen short even in this dubious department. Knuth's published error log runs to more than 900 entries, whereas here I am going to confess to only a paltry handful of mistakes. Then again, Knuth needed 10 years' work on a major software project to accumulate his budget of errors, but I was able to commit some really serious howlers in a program of a dozen lines.

Knuth remarks that keeping an error log not only helped in debugging the program but also "helped me to get to know myself." I would like to think that I too have acquired some self-knowledge from the experience of confronting my own fallibility. And it would be gratifying to suggest that by telling my story I might save others from making the same mistakes—but I don't quite believe that, and I'm not even sure it would be a good idea.

Start Spreading the News

The story begins with a loose end from my column on the Lambert W function in the March–April issue of *American Scientist*. I had been looking for a paper with the curious title "Rumours with general initial conditions," by Selma Belen and C. E. M. Pearce of the University of Adelaide, published in *The ANZIAM Journal*, which is also known as *The Australia and New Zealand Industrial and Applied Mathematics Journal*. By the time I found the paper, my column had already gone to press. This was a disappointment, because Belen and Pearce describe an illuminating

application of the W function in a context that I found interesting in its own right. Here is how they begin:

The stochastic theory of rumours, with interacting subpopulations of ignorants, spreaders and stiflers, began with the seminal paper of Daley and Kendall. The most striking result in the area—that if there is one spreader initially, then the proportion of the population never to hear the rumour converges almost surely to a proportion 0.203188 of the population size as the latter tends to infinity—was first signalled in that article. This result occurs also in the variant stochastic model of Maki and Thompson, although a typographic error has resulted in the value 0.238 being cited in a number of consequent papers.

I was intrigued and a little puzzled to learn that a rumor would die out while "almost surely" leaving a fifth of the people untouched. Why wouldn't it reach everyone eventually? And that number 0.203188, with its formidable six decimal places of precision—where did *that* come from?

I read on far enough to get the details of the models. The premise, I discovered, is that rumor-mongering is fun only if *you* know the rumor and your audience doesn't; there's no thrill in passing on stale news. In terms of the three subpopulations, people remain spreaders of a rumor as long as they continue to meet ignorants who are eager to receive it; after that, the spreaders become stiflers, who still know the rumor but have lost interest in propagating it.

The Daley-Kendall and Maki-Thompson models simplify and formalize this social process. Both models assume a thoroughly mixed population, so that people encounter each other at random, with uniform probability. Another simplifying assumption is that people always meet two-by-two, never in larger groups. The pairwise interactions are governed by a rigid set of rules:

- Whenever a spreader meets an ignorant, the ignorant becomes a spreader, while the original spreader continues spreading.
- When a spreader meets a stifter, the spreader becomes a stifter.

Brian Hayes is Senior Writer for *American Scientist*. Address: 211 Dacian Avenue, Durham, NC 27701; bhayes@amsci.org

- In the case where two spreaders meet, the models differ. In the Daley-Kendall version, both spreaders become stiflers. The Maki-Thompson rules convert only one spreader into a stifter; the other continues spreading.

- All other interactions (ignorant-ignorant, ignorant-stifler, stifler-stifler) have no effect on either party.

The rules begin to explain why rumors are self-limiting in these models. Initially, spreaders are recruited from the large reservoir of ignorants, and the rumor ripples through part of the population. But as the spreaders proliferate, they start running into one another and thereby become stiflers. Because the progression from ignorant to spreader to stifler is irreversible, it's clear the rumor must eventually die out, as all spreaders wind up as stiflers in the end. What's not so obvious is why the last spreader should disappear before the supply of ignorants is exhausted, or why the permanently clueless fraction is equal to 0.203188 of the original population.

The rumor models are closely related to well-known models of epidemic disease, where the three subpopulations are usually labeled susceptibles, infectives and removed cases. But there's a difference between rumors and epidemics. In the rumor models, it's not only the disease that's contagious but also the cure, since both spreading and stifling are communicable traits.

The Rumor Mill

I was curious to see the rumor models in action, and so I wrote a little program. I set up a population of 1,000 individuals, each of whom could be an ignorant, a spreader or a stifler. Initially there was just one spreader and all the rest were

ignorants. At the heart of the program was the following procedure, meant to implement the Daley-Kendall model (the one in which pairs of spreaders annihilate each other):

```
repeat
  choose X at random from among the
  spreaders in the population;
  choose Y at random from the entire
  population;
  if Y is an ignorant
    then make Y a spreader
  else if Y is a spreader
    then make both X and Y stiflers
  else if Y is a stifler
    then make X a stifler
until there are no more spreaders
```

When all the spreaders are gone, nothing more can change, so the program ends and reports the fraction of the population still oblivious of the rumor. This fraction, designated θ , should be 0.203188. But the result from my program, averaged over a few thousand runs, was 0.28 or 0.29—a considerable discrepancy.

At this point, let me pause to say that my big boo-boo had already been committed. Before reading on, you might want to try debugging my algorithm, or even write a program of your own.

Typos and Thinkos

Computer programming teaches humility, or at least that's my experience. In principle, the discrepancy I observed might have pointed to an error in the published result, but that wasn't my first hypothesis. I checked my own code, fully expecting to find some careless mistake—running through a loop one time too few or too many, fail-

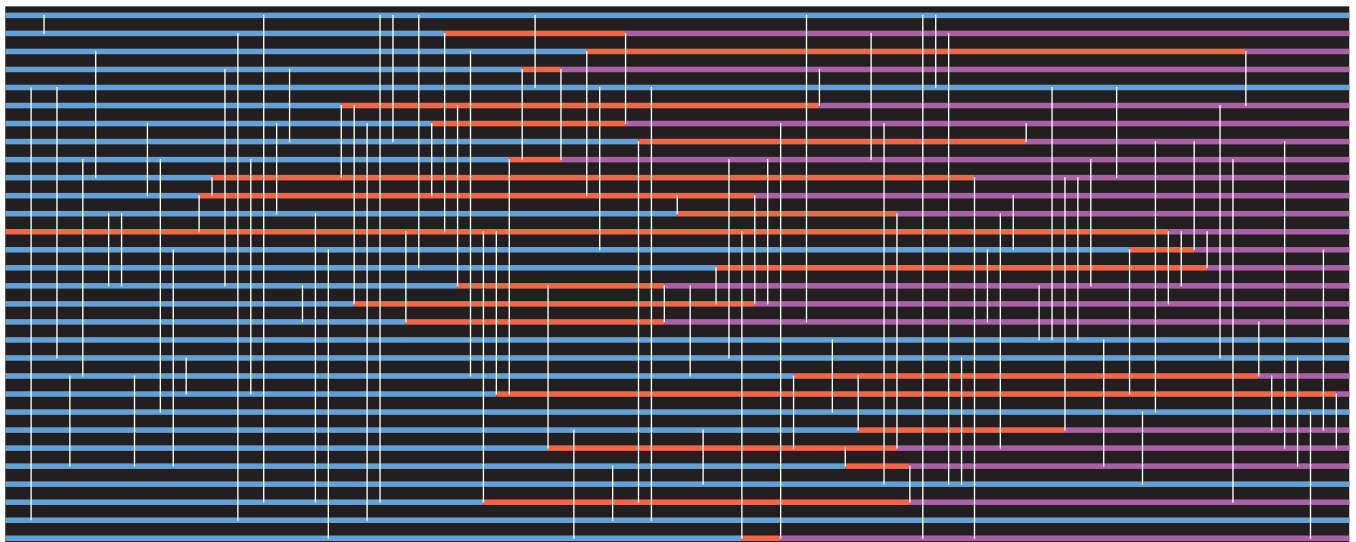


Figure 1. The birth and death of a rumor are traced in a diagram where time proceeds from left to right and each horizontal ribbon represents one member of a population. Colors encode a person's status: blue for *ignorants*, who have not heard the rumor; red for *spreaders* of the rumor; purple for *stiflers*, who know the rumor but have lost interest in it. Each white vertical link indicates an encounter between two randomly selected individuals. Spreaders communicate the rumor to ignorants, turning them into spreaders; when two spreaders meet, both become stiflers; a spreader who meets a stifler also becomes a stifler. The process begins with a single spreader in a sea of ignorants. More spreaders are recruited, but by the end they have all become stiflers. Meanwhile, 7 of the 30 individuals pass through the entire simulation without ever hearing the rumor. (This diagram and the rest of the illustrations were generated by a program that does not suffer from the error discussed in the text.)

ing to update a variable, miscalculating an array index. Nothing leapt out at me. The problem, I began to suspect, was not a typo but a thinko.

I did know of one soft spot in the program. The individuals X and Y were chosen in such a way that they could both turn out to be the same person, suggesting the strange spectacle of spreading a rumor to oneself. (“Pssst. Have I heard about...?”) When I went to fix this oddity, I discovered another bug. A variable named *spreader-count* was incremented or decremented on each passage through the loop, according to the outcome of the encounter; when this variable reached zero, the program ended. After each spreader-spreader interaction, I decreased *spreader-count* by 2—with potentially disastrous results if X and Y were identical. This was a serious flaw, which needed to be repaired; however, the change had no discernible effect on the value of θ , which remained stuck at 0.285.

I had another thought. Belen and Pearce were careful to state that their result holds only when the population size tends to infinity. Perhaps my discrepancy would go away in a larger sample. I tried a range of populations, with these results:

population	θ
10	0.354
100	0.296
1,000	0.286
10,000	0.285
100,000	0.285

The trend was in the right direction—a smaller proportion of residual ignorants as population increased—but the curve seemed to flatten out beyond 1,000, and θ looked unlikely ever to reach 0.203. Even so, it seemed worthwhile to test still larger populations, but for that I would need a faster program. I wrote a new and simpler version, dispensing with the array of individuals and merely keeping track of the number of persons in each of the three categories. With this strategy I was able to test populations up to 100 million. The value of θ remained steady at 0.285.

Looking at the distribution of θ values from single runs of the program (rather than averages over many runs) suggested another idea. Most of the results were clustered between $\theta=0.25$ and $\theta=0.35$, but there were a few outliers—runs in which 99 percent of the population never heard the rumor. I could see what must be going on. Suppose on the very first interaction X spreads the rumor to Y , and then in the second round the random selection happens to settle on X and Y again. The rumor dies in infancy, having reached only two people. Could it be that excluding these outliers would bring the average value of θ down to 0.203? I gave it a try; the answer was no.

Mea Culpa

I was stumped. I had reached the point in a debugging session where you begin to doubt your random-number generator, or even your com-

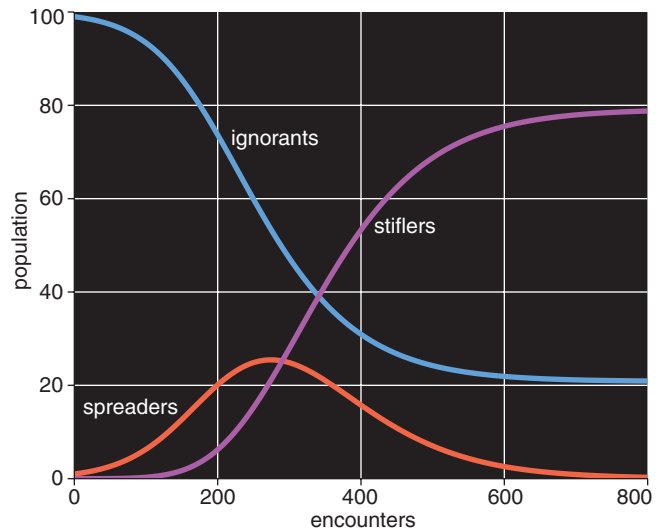


Figure 2. The dynamics of rumors are revealed by averaging the results of 100,000 runs of a model with a population of 100. Spreaders reach a peak and then die out entirely, so that the final population is divided between stiflers and ignorants. A fundamental prediction of the model is that the residual proportion of ignorants is about 0.203.

piler. As it happens, Knuth found a few compiler bugs during his work on T_EX, but for me that road has always led nowhere.

For lack of a better idea, I decided to look at the other scheme of rumor propagation, the Maki-Thompson model. As indicated above, this model differs from the Daley-Kendall one in that an encounter between two spreaders converts just one of them into a stifler. Modifying my program took only a second. When I ran it, the answer came back $\theta=0.203$. Now I was not just stumped but also thoroughly confused.

Here’s where confession becomes a test of character. There was a moment—it came in the dark of the night—when I allowed myself to entertain the notion that maybe *I* was right after all, and the rest of the world had a screw loose. I looked back at the opening paragraph of the paper by Belen and Pearce. I realized that I could make sense of it all, and reconcile their numbers with mine, merely by assuming that the Australian authors had everything umop-episdn. The number 0.203, which they identified as the result of the Daley-Kendall model, really belonged with Maki-Thompson. As for 0.238, which they called a typographical error—well, yes, that’s just what it was, a transposition of 0.283, which seemed close enough to 0.285, the value of θ I calculated for Daley-Kendall....

By morning this madness had abated, but the impasse remained. I knew I could probably settle the matter by going back to the library and looking up the sources cited by Belen and Pearce, but that seemed less than sporting. I could have tried to prove the correctness of one result or the other, but if I can’t trust myself to write a correct program, how can I be trusted to write a correct proof? Then there’s the experimental method: I might have assembled a thousand volunteers,

carefully instructed them on the Daley-Kendall rules, and set a rumor loose in their midst.

In the end, what I tried was yet another computer simulation. I decided to write a program that would mimic a real experiment as closely as possible, reproducing all the basic events of the underlying model with no shortcuts or optimizations. The image I had in mind was a crowd of people milling about like molecules in a gas, bumping into each other at random and passing on rumors during these chance collisions. This was the system I wanted to simulate.

My first program, with its explicit representation of each member of the population, was already fairly close to the goal. But I had made one refinement for the sake of computational efficiency. Because interactions in which neither party is a spreader could never affect the fate of a rumor, it seemed wasteful to include them; I avoided that waste by always choosing a spreader as the first party to an encounter. This seemed a totally harmless bit of streamlining, but now I went back and removed it. In the third version of the program, I selected two individuals at random from the entire population, checked to make sure they were not actually the same person, and then allowed them to interact according to the Daley-Kendall rules. It seemed a futile exercise, which would surely yield exactly the same result as the other programs, only slower. To my astonishment, the new program reported $\theta=0.203$.

My Bad

If you have already figured out where my reasoning went astray, I offer my congratulations. My own belated enlightenment came when I finally drew the matrix of all nine possible encounters of ignorants, spreaders and stiflers. As

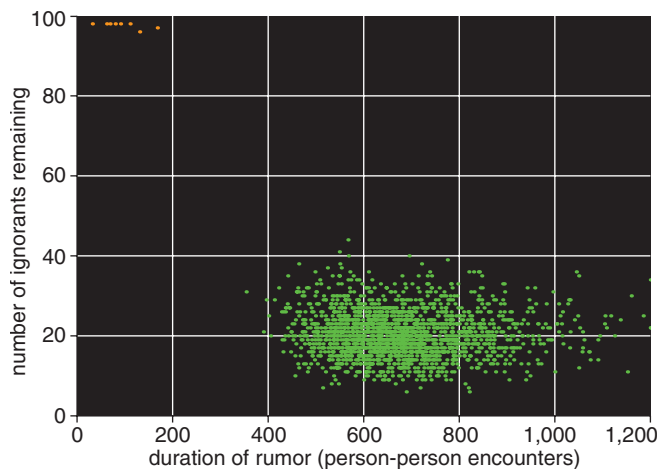


Figure 3. The longevity of a rumor and its success in permeating the population are not strongly correlated. Each dot represents a single run of the rumor program, showing both how long the rumor survived and how many people remained ignorant of it at the end. The distribution of ignorants is about the same whether the rumor dies after 500 encounters or lasts for more than 1,000. But in the upper lefthand corner are a few exceptional cases (colored orange for emphasis) of rumors that died in infancy and reached very few people.

shown in Figure 4, this diagram can serve as more than just an enumeration of possible outcomes; it encodes the entire structure and operation of the model. If we make the widths of the columns and rows proportional to the sizes of the three subpopulations, then the area of each of the nine boxes gives the probability of the corresponding two-person encounter. Choosing two participants at random is equivalent to choosing a point at random within the diagram; the outcome of the interaction is then decided by which of the nine boxes the chosen point lies within. (I am again glossing over the issue of spreading a rumor to oneself; it's a minor correction.)

To understand where I went wrong, it's enough to analyze the simplest case, where the three subpopulations are of equal size and all nine kinds of encounters have the same probability, namely $\frac{1}{9}$. The boxes at the four corners of the diagram correspond to events that do not involve a spreader and that change no one's status. Two other boxes describe ignorant-spreader encounters, which therefore have a total probability of $\frac{2}{9}$. Two more boxes correspond to spreader-stifler meetings, so those events also occur with probability $\frac{2}{9}$. But there is only one box representing spreader-spreader interactions, which accordingly must be assigned a probability of $\frac{1}{9}$. The key point is that ignorant-spreader and spreader-stifler events are each twice as likely as spreader-spreader meetings.

Now consider what happens in my first program for the Daley-Kendall model. By always choosing a spreader first, I confined all events to the middle row of the matrix, and the three boxes in this row were selected with equal probability. As a result, spreader-spreader interactions were twice as frequent as they should have been, and the rumor was extinguished prematurely.

From the point of view of probability theory, the error is an elementary one of failing to count cases properly. Perhaps a professional programmer would cite a different root cause: I had violated the old adage, "Don't start optimizing your program until you've finished writing it."

Back to the Stacks

Not all of my confusion was cleared up by the discovery of this error. In particular, the algorithm that I now knew to be incorrect for the Daley-Kendall model still seemed to give the right answer for the Maki-Thompson rules. To make sense of this situation, I finally went back to the library to find out what the original authors had said.

Daley and Kendall are Daryl J. Daley, now of the Australian National University, and David G. Kendall, a distinguished statistician and probabilist at the University of Cambridge. Their paper, published in 1965, is a model of lucid exposition, which would have spared me all my stumbling in the dark—and for that reason I'm glad I didn't see it sooner. The correct calculation of probabilities is stated very clearly (there's a factor of $\frac{1}{2}$ in

the expression for the spreader-spreader interaction). Furthermore, the origin of the mysteriously precise number 0.203188 is made plain. Those six digits come not from a discrete-event simulation like the ones I had designed but from a continuous, differential-equation version of the model. The number θ is a solution of the equation:

$$\theta e^{2(1-\theta)} = 1.$$

(This equation brings us back almost to the Lambert W function, We^W .)

Maki and Thompson are Daniel P. Maki and Maynard Thompson of Indiana University, who discussed rumors in a 1973 textbook, *Mathematical Models and Applications*. They described the rumor-passing process in terms of telephone calls, and they limited their attention to calls placed by spreaders; because of this asymmetry, only the middle row of the matrix in Figure 4 enters into the calculation, and my first program was in fact a correct implementation of their model. (At least I got *something* right.) It is almost a coincidence that Maki and Thompson arrive at the same value of θ as Daley and Kendall: Their spreader-spreader interactions are twice as likely but have only half the effect.

The paper by Belen and Pearce that launched me on this adventure also deserves a further comment. The phrase “general initial conditions” in their title refers to rumors initiated not by a single spreader but by many. One might guess that with enough spreaders, the rumor must surely permeate the entire society, but Belen and Pearce show otherwise. Measuring the fraction of those originally ignorant who remain ignorant when the rumor has finished, they find that this fraction actually increases along with the number of initial spreaders, tending to a maximum of $\frac{1}{3}$, or about 0.368. In other words, as more people spread the news, more people fail to hear it. The reason is simply that the multitude of spreaders quickly stifle one another.

By now the mathematics of rumors has acquired a vast literature. Variant models track competing rumors and counter-rumors or allow people to meet more than two at a time. Still more models study the progress of rumors through networks or lattices rather than structureless mixed populations. I have not yet had a chance to make any errors in exploring these systems.

Gnothi Seauton

Mistakes bring the gift of self-knowledge—a gift that is not always welcome. Looking back on this episode, I could summarize it as follows: I wrote a program that gave a wrong answer, and then I fiddled and fudged until I finally got the output I wanted, and then I stopped. This is not a protocol to be recommended. What’s most troubling is the uncomfortable thought that if the textbook answer had not been given to me at the outset, I would surely have been content with the result of my first, fallacious, program.

	Ign	Spr	Stf
Ign	Ign ← Ign Spr ← Spr Stf ← Stf	Ign ← Ign – 1 Spr ← Spr + 1 Stf ← Stf	Ign ← Ign Spr ← Spr Stf ← Stf
Spr	Ign ← Ign – 1 Spr ← Spr + 1 Stf ← Stf	Ign ← Ign Spr ← Spr – 2 Stf ← Stf + 2	Ign ← Ign Spr ← Spr – 1 Stf ← Stf + 1
Stf	Ign ← Ign Spr ← Spr Stf ← Stf	Ign ← Ign Spr ← Spr – 1 Stf ← Stf + 1	Ign ← Ign Spr ← Spr Stf ← Stf

Figure 4. Matrix of all possible events in the rumor model can also be taken to define the probabilities of those events. If the width of each row and column is made proportional to the number of members in the corresponding subpopulation (ignorants, spreaders and stiflers), then choosing two individuals at random is equivalent to choosing a point at random within the matrix; which of the nine boxes the point lies within determines what happens next. The three assignment statements within each box (the left arrow is usually read as “gets” or “becomes”) indicate changes in the size of each subpopulation. Note that two of the nine boxes represent ignorant-spreader interactions, and two more designate spreader-stifler encounters, but there is only one box for spreader-spreader meetings.

Still, for most of us, the only way we’ll never err is if we never try. My fellow-columnist Henry Petroski has written eloquently about the necessary role of error and failure in all worthy undertakings; as he says, falling down is part of growing up. And if we are going to make mistakes, it seems salutary to bring them out in the open and discuss their causes. Staring them in the face makes them seem a little less mortifying.

Only a little, though. A confession of this kind is not followed by absolution. And instead of “Go and err no more,” Knuth quotes Piet Hein’s advice: “Err and err and err again but less and less and less.” I take my own motto from the novelist and playwright Samuel Beckett: “Fail again. Fail better.”

Bibliography

- Belen, Selma, and C. E. M. Pearce. 2004. Rumours with general initial conditions. *The ANZIAM Journal* 45:393–400.
- Daley, D. J., and D. G. Kendall. 1965. Stochastic rumours. *Journal of the Institute of Mathematics and Its Applications* 1:42–55.
- Knuth, Donald E. 1989. The errors of T_EX. *Software—Practice & Experience* 19:607–685. Reprinted with additions in *Literate Programming*, 1992, Stanford, Calif.: Center for the Study of Language and Information, pp. 243–339.
- Maki, Daniel P., and Maynard Thompson. 1973. *Mathematical Models and Applications, with Emphasis on the Social, Life, and Management Sciences*. Englewood Cliffs, N.J.: Prentice-Hall.