

A reprint from

# American Scientist

the magazine of Sigma Xi, The Scientific Research Society

This reprint is provided for personal and noncommercial use. For any other use, please send a request Brian Hayes by electronic mail to [bhayes@amsci.org](mailto:bhayes@amsci.org).

# Fat Tails

Brian Hayes

**Y**OU'VE PROBABLY HEARD of Lake Wobegon, the little town in Minnesota where all the children are above average. There's been much head-scratching about this statistical miracle. What happens to the kids who fail to surpass themselves? Are they shipped across the lake to another little town, where all the children are below average? That practice wouldn't necessarily work to the detriment of either community. It might be like the migration from Oklahoma to California during the Dust Bowl years, which Will Rogers said raised the average intelligence of both states.

One small town that beats the law of averages is strange enough, but even more mystifying is the finding that Lake Wobegon is not unique—that in fact *everyone* is above average. In 1987 John J. Cannell, a West Virginia physician and activist, discovered that all 50 states report that their children do better than the national average on standardized tests. (And this was years before the No Child Left Behind Act!)

I can't promise to resolve these paradoxes. On the contrary, I'm going to make matters worse by describing still more funny business in the world of averages. The story that follows is about a data distribution that simply has no average. Given any finite sample drawn from the distribution, you are welcome to apply the usual algorithm for the arithmetic mean—add up the values and divide by the size of the sample—but the result won't mean much. Whatever average you calculate in this way, you can improve it just by taking a bigger sample. Perhaps this is the secret of the Lake Wobegon school board.

---

*Brian Hayes is Senior Writer for American Scientist. Additional material related to the "Computing Science" column appears in Hayes's Weblog at <http://bit-player.org>. Address: 211 Dacian Avenue, Durham, NC 27701. Internet: [bhayes@amsci.org](mailto:bhayes@amsci.org)*

*Sometimes  
the average  
is anything  
but average*

The existence of such better-than-average averages is not a new discovery; the phenomenon was already well known a century ago, and distributions with this property have become a hot topic in the past decade. Recently I stumbled upon a particularly simple illustration of the concept, and that's the story I tell here.

### Facts About Factorials

It all begins with the factorial function, a familiar item of furniture in several areas of mathematics, including combinatorics and probability theory. The factorial of a positive whole number  $n$  is the product of all the integers from 1 through  $n$  inclusive. For example, the factorial of 6 is  $1 \times 2 \times 3 \times 4 \times 5 \times 6 = 720$ .

The standard notation for the factorial of  $n$  is " $n!$ ". This use of the exclamation point was introduced in 1808 by Christian Kramp, a mathematician from Strasbourg. Not everyone is enthusiastic about it. Augustus De Morgan, an eminent British mathematician and logician, complained in 1842 that the exclamation points give "the appearance of expressing surprise and admiration that 2, 3, 4, &c. should be found in mathematical results."

One common application of the factorial function is in counting permutations, or rearrangements of things. If six people are sitting down to dinner, the number of ways they can arrange themselves at the table is  $6!$ . It's easy to see why: The first person can choose any of the six chairs, the next person

has five places available, and so on until the sixth diner is forced to take whatever seat remains.

The factorial function is notorious for its rapid rate of growth:  $10!$  is already in the millions, and  $100!$  is a number with 158 decimal digits. As  $n$  increases,  $n!$  grows faster than any polynomial function of  $n$ , such as  $n^2$  or  $n^3$ , or any simple exponential function, such as  $2^n$  or  $e^n$ . Indeed you can choose any constant  $k$ , and make it as large as you please, and there will still be some value of  $n$  beyond which  $n!$  exceeds both  $n^k$  and  $k^n$ . (On the other hand,  $n!$  grows slower than  $n^n$ .)

The steep increase in the magnitude of  $n!$  becomes an awkward annoyance when you want to explore factorials computationally. A programming language that packs integers into 32 binary digits cannot reach beyond  $12!$ , and even 64-bit arithmetic runs out of room at  $20!$ . To go further requires a language or a program library capable of handling arbitrarily large integers.

In spite of this inconvenience, the factorial function is an old favorite in computer science as well as in mathematics. Often it is the first example mentioned when introducing the concept of recursion, as in this procedure definition:

```
define f!(n)
  if n=1
    then return 1
  else return n*f!(n-1)
```

One way to understand this definition is to put yourself in the place of the procedure: You are the factorial oracle, and when someone gives you an  $n$ , you must respond with  $n!$ . Your task is easy if  $n$  happens to be 1, since calculating  $1!$  doesn't take much effort. If  $n$  is greater than 1, you may not know the answer directly, but you *do* know how to find it: just get the factorial of  $n-1$  and then multiply the result by  $n$ . Where do you find the factorial of

$n - 1$ ? Simple: Ask yourself—you're the oracle!

This self-referential style of thinking is something of an acquired taste. For those who prefer looping to recursions, here is another definition of the factorial:

```
define f!(n)
  product := 1
  for x in n downto 1
    product := product * x
  return product
```

In this case it's made explicit that we are counting down from  $n$  to 1, multiplying as we go. Of course we could just as easily count *up* from 1 to  $n$ ; the commutative law guarantees that the result will be the same. Indeed, we could arrange the  $n$  numbers in any of  $n!$  permutations. All the arrangements are mathematically equivalent, although some ways of organizing the computation are more efficient than others.

### Factoids

Playing at the keyboard one day, I came up with the following factorial-like procedure:

```
define f?(n)
  r := random(1, n)
  if r=1
    then return 1
  else return r*f?(n)
```

This might be a buggy version of a program intended to calculate factorials, but it actually does something a good deal stranger. The auxiliary procedure `random(1, n)`, invoked in the second line, is assumed to return an integer selected at random from the range 1 through  $n$ . Thus the program chooses a series of random integers, multiplies them, and stops when `random(1, n)` happens to yield a 1. It's like rolling an  $n$ -sided die and keeping a running product of all the numbers seen until a 1 appears.

Here are the results of a few sample runs of the program, for  $n=7$ :

```
4×3×2×4×7×2×1 = 1,344
2×5×4×5×4×3×6×2×2×5×1 = 288,000
7×5×5×1 = 175
4×7×6×2×6×5×6×3×5×3×3×3×7×
4×3×3×7×2×5×1 = 432,081,216,000
```

Note that  $7!$  is equal to 5,040, a value that doesn't turn up in this sample. The results that do appear are scattered over quite a wide range.

This randomized analogue of the factorial function needs a name: I shall call it the factoidal function. Also, risking the ire and ridicule of some latter-day De Morgan, I'm going to adopt the notation " $n?$ " to suggest the nondeterministic nature of the computation.

Strictly speaking, the factoidal function isn't a function—not in the mathematical sense of that word. An essential property of a function is that applying it to the same argument always yields the same value. The function call `f!(7)` will produce the value 5,040 time after time. But, as we have just seen, `f?(7)` is likely to return a different value every time it is invoked, depending on the vagaries of the random-number generator. For that matter, the procedure might not return any value at all; the program could run forever. The `f!` procedure is guaranteed to terminate, because on each recursive call (or each passage through the loop) the value of  $n$  is decremented, and eventually it has to reach 1. But `f?` halts only when the roll of a die comes up 1, which might never happen.

In practice, of course, the program always terminates, one way or another. A back-of-the-envelope calculation shows there's about a two-thirds chance that `random(1, n)` will produce a 1 in  $n$  trials. And a 1 is almost certain to turn up within  $5n$  trials. Thus if  $n$  is small (less than 1,000, say), `f?(n)` will almost surely succeed in calculating a value. If  $n$  is very large, the product being computed is likely to fill up all available memory, and the program will terminate with an error message.

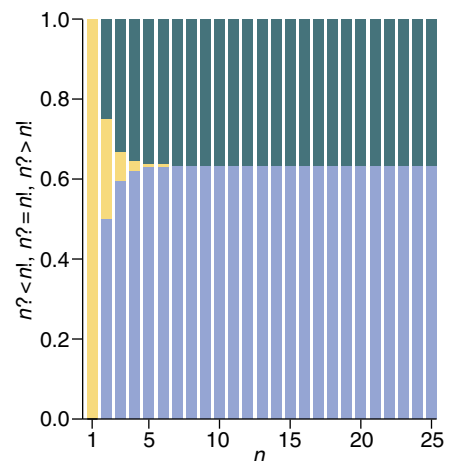
Because of the element of randomness in the factoidal definition, it makes no sense to ask about *the* value of  $n?$ . The best we can hope for is to understand the statistical distribution of values. As a rule, this would mean estimating quantities such as the average value and the variance or standard deviation. But those familiar statistical tools are problematic in this case, so let's start by asking an easier question: How do factoidals compare with factorials? Is  $n?$  usually larger than  $n!$ , or smaller? (They can also be equal, of course, but as  $n$  increases, the probability of such an exact match goes to zero.)

When I first began puzzling over this question, I convinced myself that  $n?$  would usually exceed  $n!$ . Mercifully, I have forgotten the details of the "reasoning" that led me to this conclusion; it had something to do with the

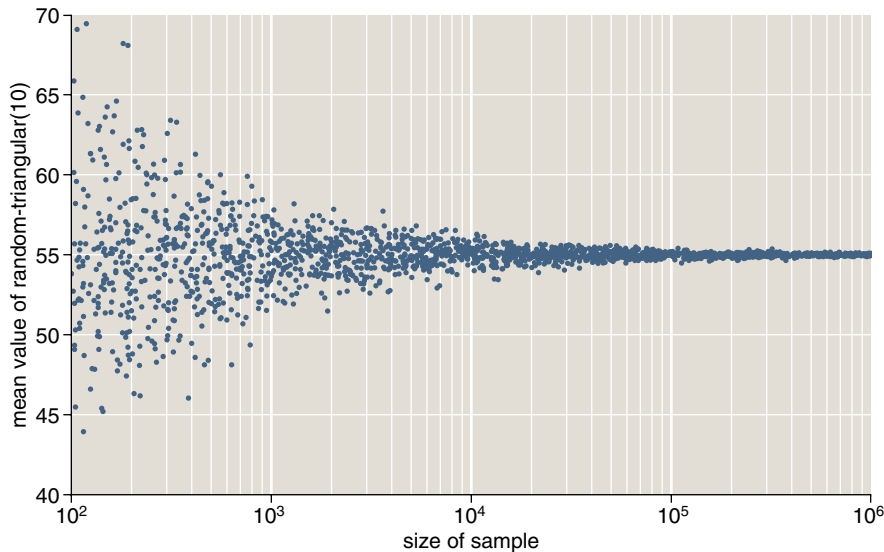
idea that only finitely many possible values of  $n?$  are less than  $n!$ , whereas infinitely many are greater. There's no point in reconstructing my argument, because it was totally wrong. When I wrote a program to answer the question experimentally, it became clear that nearly two-thirds of the  $n?$  values are less than  $n!$ . (The results appear in the illustration below.)

You can get an inkling of what's going on here by looking at a small example, say  $n=3$ , and counting the cases in which the final product is less than  $3!$ , or 6. For  $n=3$  the only possible outputs of the random number generator are 1, 2 and 3, each of which appears with probability  $\frac{1}{3}$ . The simplest event is that a 1 comes up on the first try, in which case the product is obviously less than 6; this happens with probability  $\frac{1}{3}$ . There are two possible sequences of just two factors, namely a 2 followed by a 1 and a 3 followed by a 1; each of these outcomes has a probability of  $(\frac{1}{3})^2$ , or  $\frac{1}{9}$ , and again the products are less than 6. A pencil-and-paper enumeration shows that there is only one other sequence of factors whose product is less than 6, namely 2, 2, 1; here the probability is  $\frac{1}{27}$ . The sum of these probabilities is  $\frac{1}{3} + \frac{2}{9} + \frac{1}{27} = \frac{16}{27}$ , or 0.5926. This is in agreement with the experimental results.

As  $n$  increases, the proportion of  $n?$  values less than  $n!$  converges on the value 0.6322, and the proportion that are greater tends toward 0.3678. Is there anything special about these



Values produced by the random factoidal function  $n?$  are compared with the corresponding factorial  $n!$ . For each value of  $n$ , the bars show the proportion of  $n?$  values that are smaller than  $n!$  (blue), equal to  $n!$  (yellow) and greater than  $n!$  (green). For large  $n$ , the proportion of  $n?$  values greater than  $n!$  is 0.3678, which is equal to  $1/e$ .

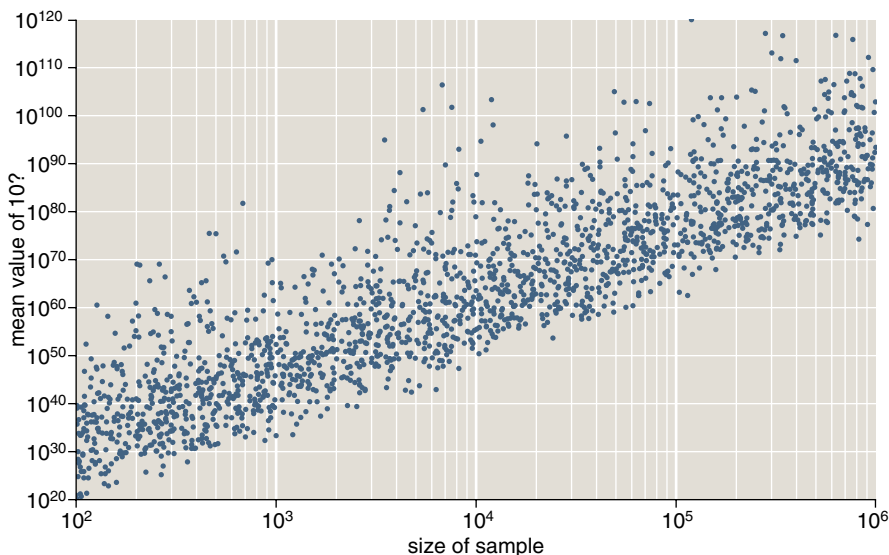


The arithmetic mean is well-defined for an additive analogue of the factoidal function—a randomized computation of triangular numbers. The algorithm selects integers randomly in the range 1 through 10, summing the numbers seen until the first 1 is drawn. The process is repeated, and the mean value is calculated. For each of the 2,000 dots shown here, vertical position indicates the calculated mean, and horizontal position gives the sample size. With increasing sample size, the sample means converge to an overall mean of 55, which is the 10th triangular number.

numbers, and can we explain why this particular proportion should crop up? I think so. Note that on each iteration, the probability of getting a 1 and thereby ending the factoidal process is  $1/n$ . That means the probability of getting anything other than a 1, and continuing the sequence of factors, is  $(n-1)/n$ . Then the probability of avoiding 1 twice in a row is  $((n-1)/n)^2$ , the probability of going on to three

factors is  $((n-1)/n)^3$ , and so on. To have a high likelihood that  $n?$  will exceed  $n!$ , the chain of factors has to be extended to a length of at least  $n$ . The probability of reaching this point is  $((n-1)/n)^n$ ; as  $n$  increases, this expression converges to  $1/e$ , with a numerical value of 0.3678—just what is observed.

If most values of  $n?$  are less than  $n!$ , then the average of all  $n?$  values should also be less than  $n!$ , shouldn't it? We'll



The arithmetic mean is undefined for the factoidal function. Here numbers are chosen at random from the range 1 through 10 and multiplied until a 1 is drawn. Each of the 2,000 dots is positioned according to the sample size and the calculated mean for that sample. As the sample size increases, the calculated mean does not converge to a stable value; on the contrary, it rises exponentially, and so does the dispersion of observed mean values. For samples of about 1,000 values, the mean is near  $10^{50}$ ; for samples of size 100,000, it is 30 orders of magnitude greater.

see. But first another digression, to look at a closely related function that shows how averaging is *supposed* to work.

### A Well-Mannered Function

If you delve into the code for the factorial function and replace the multiplication sign with a plus sign, you wind up with a procedure for calculating triangular numbers—1, 3, 6, 10, 15, 21.... To see why they're called triangular, think of 10 bowling pins or 15 billiard balls. Whereas the factorial of 4 is  $1 \times 2 \times 3 \times 4 = 24$ , the corresponding triangular number is  $1 + 2 + 3 + 4 = 10$ . (There is a well-known shortcut for computing the  $n$ th triangular number:  $n(n+1)/2$ . It's interesting that no comparable shortcut exists for factorials, although there are approximations.)

Having converted the  $n!$  code into a program for generating triangular numbers, we can clearly make the same change in the program for  $n?$ . The result will be a procedure that rolls an  $n$ -sided die and keeps a running sum (rather than a product) until a 1 turns up.

Sums of random variables are much better-behaved than products. With randomized triangular sums, the algorithm for calculating the arithmetic mean works flawlessly. Generate a sample of values (all for the same  $n$ ), add them up, divide by the size of the sample, and you get an estimate of the mean. With a small sample, the estimate is somewhat unreliable, so that repeating the procedure is likely to produce a substantially different result. But as the sample size increases, the estimates grow more consistent. The illustration at the top of this page shows the convergence of sample means toward the true mean for 2,000 samples of various sizes.

### By No Means

How different are the statistics of the factoidal process! When I first began playing with the  $n?$  function, I was curious about its average value, and so I did a quick computation with a small sample—100 repetitions of  $10?$ . The result that came back was much larger than I had expected, in the neighborhood of  $10^{25}$ . When I repeated the computation several times, I continued to get enormous numbers, and furthermore they were scattered over a vast range, from less than  $10^{20}$  to well over  $10^{30}$ . The obvious strategy was to try a larger sample in order to smooth out

the fluctuations. But when I averaged factoidals 10,000 at a time, and then a million at a time, the numbers got even bigger, and the variations wider.

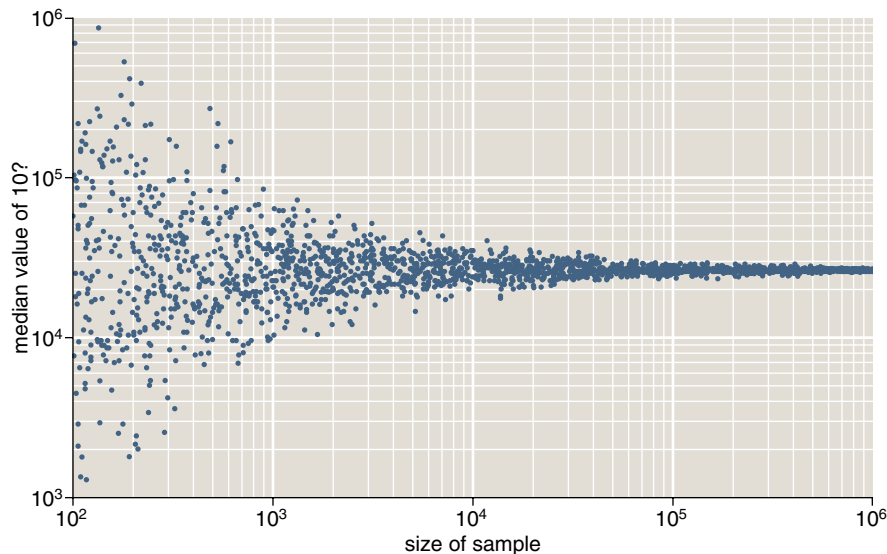
The illustration at the bottom of page 202 shows what I was up against. Each dot represents a sample of runs of the  $n?$  program; a dot's horizontal position indicates the sample size, and its vertical position gives the arithmetic mean calculated from that sample. In all cases the value of  $n$  is 10. It's important to emphasize that this is *not* a graph of  $n?$  as a function of  $n$ ; the value of  $n$  is fixed. All that changes in moving from left to right across the graph is the size of the sample over which the average is computed. There is no sign of convergence here. The trend is continuously upward: The more trials in the sample, the larger the calculated mean. And because both scales in the graph are logarithmic, the apparent straight-line trend in the mean actually represents exponential growth. The "average" value of  $10?$  is somewhere near  $10^{40}$  or  $10^{50}$  if you average over 1,000 trials, but it rises to roughly  $10^{90}$  if you go on to collect a million samples. (For comparison,  $10!$  is roughly  $10^6$ —or more precisely 3,628,800.)

The dispersion of the dots around the trend line also shows no sign of diminishing as the sample size increases. Thus the variance or standard deviation of the data is also impossible to pin down.

Odd, isn't it? Generally, if you are conducting an experiment, or making a measurement, or taking an opinion survey, you expect that collecting more data will yield greater accuracy and consistency. Here, more data just seems to make a bad situation worse.

With a closer look at the factoidal data, it's not hard to understand what's going wrong with the computation of the mean. Although the majority of  $10?$  values are comparatively small (less than 3,628,800), every now and then the factoidal process generates an enormous product—a rogue, a monster. The larger the sample, the greater the chance that one of these outliers will be included. And they totally dominate the averaging process. If a sample of 1,000 values happens to include one with a magnitude of  $10^{100}$ , then even if all the rest of the data points were zero, the average would still be  $10^{97}$ .

The arithmetic mean is not the only tool available for characterizing what



The median of the factoidal process is more informative than the arithmetic mean. The median is the midpoint value of a sample—the value chosen such that half the other values are smaller and half are larger. For data generated by  $10?$ , the median converges to a value of about 27,000. Note that this is much smaller than  $10!$ , which is equal to 3,628,800.

statisticians call the central tendency of a data set. There is also the geometric mean. For two numbers  $a$  and  $b$ , the geometric mean is defined as the square root of  $a \times b$ ; more generally, the geometric mean of  $k$  numbers is the  $k$ th root of their product. The geometric mean of samples taken from the factoidal process suffers from none of the problems encountered with the arithmetic mean. It converges, though somewhat slowly, to a stable value. Moreover, it turns out that the geometric mean of  $n?$  is simply  $n!$ , so this is a highly informative measure. Perhaps it should not be a surprise that factoidals are better described by a statistic based on multiplication than by one based on addition.

The median of  $n?$  is also well-defined. The median is the midpoint value of a data set—the item that is greater than half the others and less than half. Because it merely counts the number of greater and lesser values, without considering their actual magnitudes, it is insensitive to the outliers that cause havoc with the arithmetic mean. For samples of  $10?$  the median converges on a value near 27,000 (notably smaller than  $10!$ ).

Still another way to tame the factoidal is to take logarithms. If you determine the logarithm of each  $n?$  value, then calculate the arithmetic mean of the logarithms, the result converges very nicely. (Note that taking the mean of the logarithms is not the same as taking the logarithm of the means.)

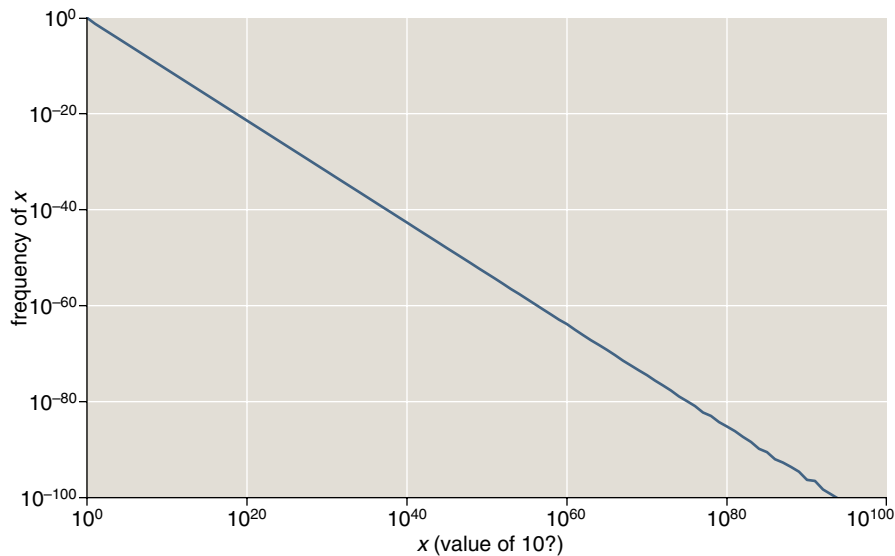
The success of this strategy does not come as a surprise. Logarithms reduce multiplication to addition. Essentially, then, taking the logarithm of the  $n?$  values converts the factoidal process into the corresponding triangular-number calculation. Logarithms are also at work behind the scenes in computing the geometric mean.

Even with other statistical methods available, it's disconcerting to face the failure of something so familiar and elementary and ingrained as the arithmetic mean. It's like stumbling into an area of mathematics where Euclid's parallel postulate no longer applies, or the commutative law has been repealed. To be sure, such areas exist, and exploring them has enriched mathematics. Distributions without a mean or variance have likewise broadened the horizons of statistics. All the same, they take some getting used to.

### Fat Tails

The little procedure I have named the factoidal function is so simple that I'm sure someone must have noticed it before. I have not found a mention of this specific process, but slightly more general models involving products of random numbers do appear in the literature. (Review articles by Mark Newman of the University of Michigan and by Michael Mitzenmacher of Harvard University are particularly helpful.)

The context of these discussions is the study of heavy-tailed or fat-tailed distributions. The familiar normal distribu-



The distribution of factoidal values follows a straight line on logarithmic scales, a telltale sign of a power-law behavior. For each possible value,  $x$ , of the function  $10^n$ , the graph records the relative frequency at which that value of  $x$  is observed. The frequency of  $x$  is proportional to  $x^{-\alpha}$ , where  $\alpha$  has a value of about 1.07. The bumpiness at the extreme tail of the distribution can be attributed to the finite sample size (10 million evaluations of the  $n?$  function).

tion is *not* in this class: It is lean-tailed. The extremes of the normal probability curve, far from the peak, fall away exponentially, so that unlikely events become *really* unlikely and are never seen. Fat-tailed distributions decay more slowly, allowing room for outliers and freaks. Human height is a normally distributed variable; most people are less than two meters tall, and nobody reaches three meters. Human wealth has a fat-tailed distribution; worldwide, median net worth is a little over \$2,000, but there are also millionaires and billionaires. (If height had the same distribution as wealth, there would be people two million meters tall.)

The distribution of wealth was one of the subjects that first aroused interest in fat-tailed distributions, starting with the work of the Italian economist Vilfredo Pareto in the 1890s. Later it emerged that word frequencies in natural languages are also described by a fat-tailed distribution, usually called Zipf's law, after George Kingsley Zipf. The sizes of cities offer another example: If urban populations were normally distributed, we wouldn't have Mumbai or São Paulo. In the past decade or so, it seems like fat tails have been turning up everywhere: in the number of links to Web sites and citations of scientific papers, in the fluctuations of stock-market prices, in the sizes of computer files.

The classic fat-tailed distribution is one where the decay of the tails is

described by a power law. The probability of observing some quantity  $x$  goes as  $x^{-\alpha}$ , where  $\alpha$  is a constant; the smaller the value of  $\alpha$ , the fatter the tails. When  $\alpha$  is less than 2, the mean of the distribution does not exist. Drawn on a graph with logarithmic scales, a power-law distribution takes the form of a straight line. Another fat-tailed distribution, called the lognormal, follows a straight line over a certain range but at some point takes a sudden nosedive. The lognormal, as the name suggests, is the distribution formed by variables whose logarithms are normally distributed.

What about the factoidal function—which distribution describes the  $n?$  values? My first guess was a lognormal, based on a vague intuition that the logarithms of the  $n?$  products should indeed be normal. So much for *my* intuition! A log-log graph of the factoidal function shows clear evidence of power-law behavior: The graph is a straight line, with no hint of the “bended knee” to be expected in a lognormal. The calculated value of the exponent  $\alpha$  is about 1.07, well inside the range where the mean and variance cease to exist.

With guidance from Newman and Mitzenmacher I eventually came to understand why the factoidal follows a power law. They pointed me to a paper by William J. Reed of the University of Victoria in Canada and Barry D. Hughes of the University

of Melbourne in Australia. Reed and Hughes show that when a process of exponential growth is stopped at random times, the resulting distribution of values follows a power law. One of their examples is multiplication of random numbers with mean  $\mu$ , stopped after a random number of terms. The factoidal function is merely a special case of this process.

The shape of a probability distribution can have grave consequences in many areas of life. If the size and intensity of hurricanes follows a normal distribution, we can probably cope with the worst of them; if there are monster storms lurking in the tail of the distribution, the prospects are quite different. Those who make a profession of risk assessment—insurance underwriters, financial analysts—take a keen interest in these questions.

Could the fat-tails phenomenon clear up the Lake Wobegon mystery? Well, maybe it can teach us a new way to understand the phrase, “All the children are above average.” It's not about drawing a line through the population and having each and every child above the line. The sense of “all” is not “each and every” but the totality of children. If we can believe that human talents and abilities have some distribution that escapes the bounds of means and variances, then “all” children can indeed be above average.

### Bibliography

- Adler, Robert J., Raisa E. Feldman and Murad S. Taqqu (editors). 1998. *A Practical Guide to Heavy Tails: Statistical Techniques and Applications*. Boston: Birkhäuser.
- Cannell, John Jacob. 1988. Nationally normed elementary achievement testing in America's public schools: How all 50 states are above the national average. *Educational Measurement: Issues and Practice* 7(2):5–9.
- Gabaix, Xavier. 1999. Zipf's law for cities: an explanation. *The Quarterly Journal of Economics* 114(3):739–767.
- Keillor, Garrison. 1985. *Lake Wobegon Days*. New York: Viking.
- Mitzenmacher, Michael. 2004. A brief history of generative models for power law and lognormal distributions. *Internet Mathematics* 1(2):226–251.
- Newman, M. E. J. 2005. Power laws, Pareto distributions and Zipf's law. *Contemporary Physics* 46:323–351.
- Reed, William J., and Barry D. Hughes. 2002. From gene families and genera to incomes and internet file sizes: Why power laws are so common in nature. *Physical Review E* 66(6):067103.
- Sigman, Karl. 1999. A primer on heavy-tailed distributions. *Queueing Systems: Theory and Applications* 33(1–3):261–275.