CHAPTER

34

Explaining Benford's Law

Digital Signal Processing usually involves signals with either *time* or *space* as the independent parameter, such as audio and images, respectively. However, the power of DSP can also be applied to signals represented in other domains. This chapter provides an example of this, where the independent parameter is the *number line*. The particular example we will use is Benford's Law, a mathematical puzzle that has caused people to scratch their heads for decades. The techniques of signal processing provide an elegant solution to this problem, succeeding where other mathematical approaches have failed.

Frank Benford's Discovery

Frank Benford was a research physicist at General Electric in the 1930s when he noticed something unusual about a book of logarithmic tables. The first pages showed more wear than the last pages, indicating that numbers beginning with the digit 1 were being looked up more often than numbers beginning with 2 through 9. Benford seized upon this idea and spent years collecting data to show that this pattern was widespread in nature. In 1938, Benford published his results, citing more than 20,000 values such as atomic weights, numbers in magazine articles, baseball statistics, and the areas of rivers.

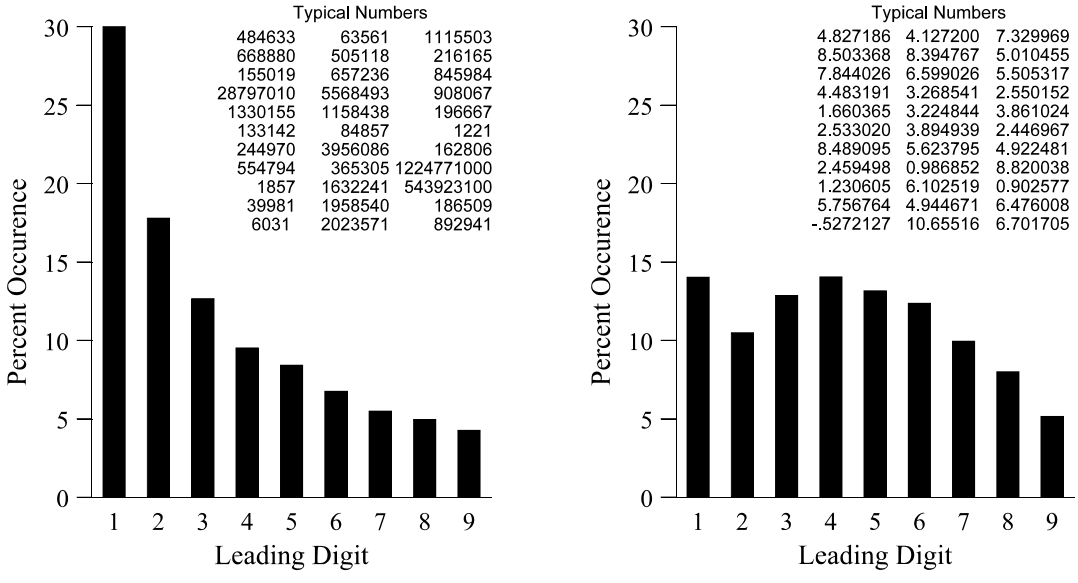
This pattern of numbers is unexpected and counterintuitive. In fact, many do not believe it is real until they conduct an experiment for themselves. I didn't! For instance, go through several pages of today's newspaper and examine the **leading digit** of each number. That is, start from the left of each number and ignore the sign, the decimal point and any zeros. The first digit you come to, between 1 and 9, is the leading digit. For example, 3 is the leading digit of 37.3447, and 6 is the leading digit of -0.06345. Since there are nine possible digits, you would expect that one-ninth (11.11%) of the numbers would have 1 in the leading digit position. However, this is not what you will find– about 30.1% of the numbers will start with 1. It gets even stranger from here.

FIGURE 34-1
Frank Albert Benford, Jr., (1883-1948) was an American electrical engineer and physicist. In 1938 he published a paper entitled "The Law of Anomalous Numbers." This is now commonly called Benford's Law.

Figure 34-2 shows two examples of Benford's law. The histogram on the left is for 14,414 numbers taken from the income tax returns of U.S. corporations. The pattern here is obvious and very repeatable. The leading digit in these numbers is a 1 about 30.1% of the time, a 2 about 17.6% of the time, and so on. Mathematicians immediately recognize that these values correspond to the spacing on the logarithmic number line. That is, the distance between 1 and 2 on the log scale is $log(2) - log(1) = 0.301$. The distance between 2 and 3 is $log(3) - log(2) = 0.176$, and so on. Benford showed us that this logarithmic pattern of leading digits is extremely common in nature and human activities. In fact, even the physical constants of the universe follow this pattern– just look at the tables in a physics textbook.

On the other hand, not all sets of numbers follow Benford's law. For example, the histogram in Fig. 34-2b was generated by taking a large number of samples from a computer random number generator. These particular numbers follow a normal distribution with a mean of five and a standard deviation of three. Changing any of these parameters will drastically change the shape of this histogram, with little apparent rhyme or reason. Obviously, these numbers do not follow the logarithmic leading-digit distribution. Likewise, most of the common distributions you learned about in statistics classes do not follow Benford's law. One of the primary mysteries of Benford's law has been this seemingly unpredictable behavior. Why does one set of numbers follow the logarithmic pattern, while another set of numbers does not?

As if this wasn't mysterious enough, Benford's law has another property that is certain to keep you up at night. Figure 34-2a was created from numbers that appear in U.S. tax returns, and therefore each of these numbers is a dollar value. But what is so special about the U.S. dollar? Suppose that you are a financial expert in India and want to examine this set of data. To make it easier you convert all of the dollar values to Indian rupees by multiplying each number by the current conversion rate. It is likely that the leading digit of all 14,414 numbers will be changed



a. Tax return numbers, *Benford's law* b. RNG numbers, *not Benford's law*

FIGURE 34-2
Two examples of leading-digit histograms. The left figure shows the leading-digit distribution for 14,414 numbers taken from U.S. Federal income tax returns. The figure on the right is for numbers produced by a computer random number generator (RNG). This shows one of the longstanding mysteries of Benford's law– Why do some sets of numbers follow the law (such as tax returns), while others (such as this RNG) do not? Many have claimed that this is some sort of secret code hidden in the fabric of Nature.

by this conversion. Nevertheless, about 30.1% of the converted numbers will still have a leading digit of 1. In other words, if a set of numbers follows Benford's law, multiplying the numbers by any possible constant will create another set of numbers that also follows Benford's law. A system that remains unchanged when multiplied by a constant is called **scale invariant**. Specifically, groups of numbers that follow Benford's law are scale invariant. Likewise, groups of numbers that do not follow Benford's law are not. For instance, this procedure would scramble the shape of the histogram in Fig. 34-2b.

Now suppose that this tax return data is being examined by an alien from another planet. Since he has eight fingers, he converts all of his numbers to base 8. Like before, most or all of the leading digits will change in this procedure. In spite of this, the new group of numbers also follows Benford's law (taking into account that there are no 8's or 9's in base 8). This property is called **base invariance**. In general, if a group of numbers follows Benford's law in one base, it will also follow Benford's law if converted to another base. However, there are some exceptions to this that we will look at later.

What does this all mean? Over the last seven decades Benford's law has achieved almost a cult following. It has been widely claimed to be evidence of some mysterious or paranormal property of our universe. For instance, Benford himself tried to connect the mathematics with Nature, claiming that *mere Man counts arithmetically, 1,2,3,4..., while Nature counts* e^0, e^x, e^{2x}, e^{3x}, *and so on.* In another popular version, suppose that nature contains some underlying and universal distribution of numbers. Since it is universal, it should look the same regardless of how we choose to examine it. In particular, it should not make any difference what *units* we associate with the numbers. The distribution should appear the same if we express it in dollars or rupees, feet or meters, Fahrenheit or Celsius, and so on. Likewise, the appearance should not change when we examine the numbers in different bases. It has been mathematically proven that the logarithmic leading-digit pattern is the only distribution that fulfils these *invariance* requirements. Therefore, if there is an underlying universal distribution, Benford's law must be it. Based on this logic, it is very common to hear that Benford's law only applies to numbers that have *units* associated with them. On the other end of the spectrum, crackpots abound that associate Benford's law with psychic and other paranormal claims.

Don't waste your time trying to understand the above ideas; they are completely on the wrong track. There is no "universal distribution" and this phenomenon is unrelated to "units." In the end, we will find that Benford's law looks more like a well-executed magic trick than a hidden property of the universe.

Homomorphic Processing

Enjoy learning about Benford's law, but don't lose sight of the purpose of this chapter. Focus on the overall method:

> "*If the tool you have is a hammer,*
> *make the problem look like a nail."*

In DSP this approach is called **homomorphic processing**, meaning "the same structure." In science and engineering it is common to encounter signals that are difficult to understand or analyze. The strategy of homomorphic processing is to convert this unmanageable situation into a conventional linear system, where the analysis techniques are well understood. This is done by applying whatever mathematical transforms or tricks are needed for the particular application.

For instance, the classic use of homomorphic processing is to separate signals that have been multiplied, such as: $a(t) = b(t) \times c(t)$. This can be converted into a linear system, i.e., signals that are added together, by taking the logarithm: $\log[a(t)] = \log[b(t)] + \log[c(t)]$. Notice that this is taking the log of the dependent parameter. In our analysis of Benford's law we will take the log of the independent parameter. Two different

techniques to keep in your bag of DSP tricks. In the next section several other tricks will be presented, such as inventing the *Ones Scaling Test*, and evoking a *sampling function*.

It this sounds complicated, you're right; it certainly can be. There is no guarantee that it is even possible to convert an arbitrary problem into the form of a linear system. Even if it is possible, it may require a series of nasty steps that take considerable time to develop. However, if you are successful in applying the homomorphic approach the rewards will immediately flow. You can say goodbye to a difficult problem, and hello to a representation that is simple and straightforward.

The following analysis of Benford's law is conducted in three steps. In step one we will define a statistical procedure for determining how well a set of numbers follows Benford's law, called the *Ones Scaling Test*. In step two we will move from statistics to probability, expressing the problem in the form of a convolution. In step three we use the Fourier Transform to solve the convolution, giving us the explanation we are looking for.

The Ones Scaling Test

Given a set of numbers, the simplest test for Benford's law is to count how many of the numbers have 1 as the leading digit. This fraction will be about 0.301 if Benford's law is being followed. However, even finding this value exactly is not sufficient to conclude that the numbers are obeying the law. For instance, the set might have 30.1% of the numbers with a value of 1.00, and 69.9% with a value of 2.00. We can overcome this problem by including a test for *scale invariance*. That is, we multiply each number in the set by some constant, and then recounting how many numbers have 1 as their leading digit. If Benford's law is truly being followed, the percentage of numbers beginning with the digit 1 will remain about 30.1%, regardless of the constant we use.

A computer program can make this procedure more systematic, such as the example in Table 34-1. This program loops through the evaluation 696 times, with each loop multiplying all numbers in the group by 1.01. On the first loop each of the original numbers will be multiplied by 1.01. On the second loop each number will be multiplied by 1.01 again, in addition to the multiplication that took place on the first loop. By the time we reach the 80^{th} loop, each number will have been multiplied by 1.01 a total of 80 times. Therefore, the numbers on the 80^{th} loop are the same as multiplying each of the original numbers by 1.01^{80}, or 2.217. At the completion of the program the numbers will have been multiplied 696 times, equivalent to multiplying the original numbers by a constant of $1.01^{696} \approx 1{,}000$. In other words, this computer program systematically scales the data in small increments over about three orders of magnitude.

The fraction of numbers having 1 as the leading digit is tallied on each of these 696 steps and stored in an array, which we will call the **Ones Scaling Test**. Figure 34-3 shows the values in this array for the two

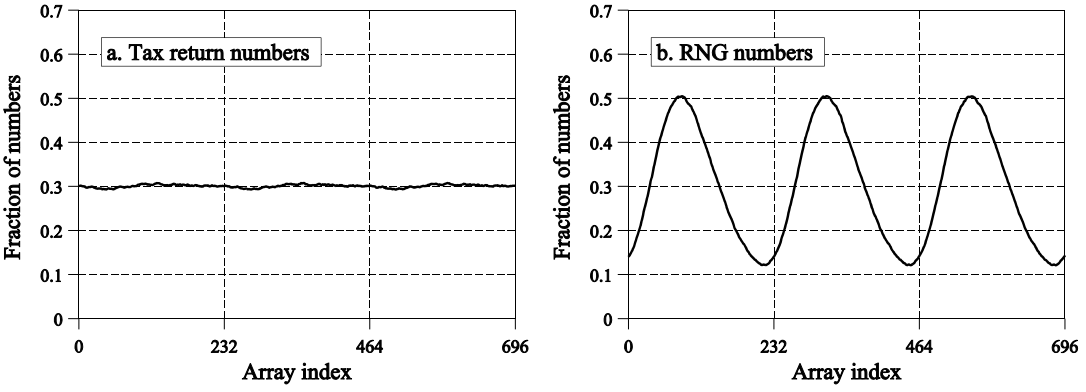


FIGURE 34-3
The Ones Scaling Test for the examples in Fig. 34-2. The Ones Scaling Test determines the fraction of numbers having a leading digit of one, as the set of number is repeatedly multiplied by a constant slightly greater than unity, such as 1.01. If the set of numbers follows Benford's law, the fraction will remain close to 0.301, as shown in (a). The fraction departing from 0.301 proves that the numbers do not follow Benford's law, such as in (b).

examples in Fig. 34-2. As expected, the Ones Scaling Test for the income tax numbers is a relatively constant value around 30.1%, proving that it follows Benford's law very closely. As also expected, the Ones Scaling Test for the random number generator shows wild fluctuations, as high as 51% and as low as 12%.

An important point to notice in Fig. 34-3 is that the Ones Scaling Test is *periodic*, repeating itself when the multiplication constant reaches a factor of ten. In this example the period is 232 entries in the array, since $1.01^{232} \approx 10$. Say you start with the number 3.12345 and multiply it by 10 to get 31.2345. These two numbers, 3.12345 and 31.2345, are exactly the same when you are only concerned with the leading digit, and the entire pattern repeats.

Pay particular attention to the operations in lines 400 to 430 of Table 34-1. This is where the program determines the leading digit of the number being evaluated. In line 310, one of the 10,000 numbers being tested is transferred to the variable: *TESTX*. The leading digit of *TESTX*, eventually held in the variable *LD*, is calculated in four steps. In line 400 we eliminate the sign of the number by taking the absolute value. Lines 410 and 420 repeatedly multiply or divide the number by a factor a ten, as needed, until the number is between 1 and 9.999999. For instance, line 410 tests the number for being less than 1. If it is, the number is multiplied by 10, and the line is repeated. When the number finally exceeds 1, the program moves to the next line. In line 430 we extract the integer portion of the number, which is the leading digit. Make sure you understand these steps; they are key to understanding what is really going on in Benford's law.

```
100 ' INVESTIGATING BENFORD'S LAW: THE ONES SCALING TEST
110 '
120 '                                   'DIMENSION THE ARRAYS
130 DIM OST(696)                          'The "Ones Scaling Test" array.
140 DIM X(9999)                           'The 10,000 numbers being tested.
150 '
160 FOR I = 0 TO 9999                   'GENERATE 10,000 NUMBERS FOR TESTING
170   X(I) = RND                          ' RND returns a random number uniformly
180 NEXT I                                '   distributed between 0 and 1.
190 '
200 '                                   'CALCULATE THE ONE SCALING TEST ARRAY
210 FOR K = 0 to 696                      'Loop for each entry in the OST array.
220   NRONES = 0                          'NRONES counts how many leading digits are one.
230   '
300   FOR I = 0 TO 9999                   'Loop through all 10,000 numbers being tested.
310     TESTX = X(I)                      'Load number being tested into variable, TESTX.
320     '
330     '                                 'Find the leading digit, LD, of TESTX.
400     TESTX = ABS(TESTX)
410     IF TESTX <   1 THEN TESTX = TESTX * 10: GOTO 410
420     IF TESTX >= 10 THEN TESTX = TESTX / 10: GOTO 420
430     LD = INT(TESTX)
440     '
500     '                                 'If leading digit is 1, increment counter.
510     IF LD = 1 THEN NRONES = NRONES + 1
520   NEXT I
530   '
540   OST(K) = NRONES / 10000             'Store the calculated fraction in the array.
550   '
600   FOR I = 0 TO 9999                   'Multiply test numbers by 1.01, for next loop.
610     X(I) = X(I) * 1.01
620   NEXT I
630   '
700 NEXT K
710 '                                     'The Ones Scaling Test now resides in OST(  ).
```

TABLE 34-1

Writing Benford's Law as a Convolution

The previous section describes the Ones Scaling Test in terms of **statistics**, i.e., the analysis of actual numbers. Our task now is to rewrite this test in terms of **probability**, the underlying mathematics that govern *how* the numbers are generated.

As discussed in Chapter 2, the mathematical description of a process that generates numbers is called the **probability density function**, or **pdf**. In general, there are two ways that the shape of a particular pdf can be known. First, we can understand the physical process that generates the numbers. For instance, the random number generator of a computer falls in this category. We know what the pdf is, because it was specifically designed to have this pdf by the programer that developed the routine.

Second, we can estimate the pdf by examining the generated values. The income tax return numbers are an example of this. It seems unlikely that anyone could mathematically understand or predict the pdf of these numbers; the processes involved are just too complicated. However, we can take a large group of these numbers and form a histogram of their values. This histogram gives us an estimate of the underlying pdf, but isn't exact because of random statistical variations. As the number of samples in the histogram becomes larger, and the width of the bins is made smaller, the accuracy of the estimate becomes better.

The statistical version of the Ones Scaling Test analyzes a group of numbers. Moving into the world of probability, we will replace this group of numbers with its probability density function. The pdf we will use as an example is shown in Fig. 34-4a. The mathematical name we will give this example curve is **pdf(g)**. However, there is an important catch here; we are representing this probability density function along the *base-ten logarithmic number line*, rather than the conventional linear number line. The position along the logarithmic axis will be denoted by the variable, **g**. For instance, g = -2 corresponds to a value of 0.01 on the linear scale, since log(0.01) = -2. Likewise, g = 0 corresponds to 1, g = 1 corresponds to 10, and so on.

Many science and engineering graphs are presented with a logarithmic x-axis, so this probably isn't a new concept for you. However, a special problem arises when converting a probability density function from the linear to the logarithmic number line. The usual way of moving between these domains is simple point-to-point mapping. That is, whatever value is at 0.01 on the linear scale becomes the value at -2 on the log scale; whatever value is at 10 on the linear scale becomes the value at 1 on the log scale, and so on. However, the pdf has a special property that must be taken into account. For instance, suppose we know the shape of a pdf and want to determine how many of the numbers it generates are greater than 3 but less than 4. From basic probability, this fraction is equal to the area under the curve between the values of 3 and 4. Now look at what happens in a point-to-point mapping. The locations of 3 and 4 on the linear scale become log(3) = 0.477 and log(4) = 0.602, respectively, on the log scale. That is, the distance between the two points is 1.00 on the linear scale, but only 0.125 on the logarithmic number line. This changes the area under the curve between the two points, which is simply not acceptable for a pdf.

Fortunately, this is quite simple to correct. First, transfer the pdf from the linear scale to the log scale by using a point-to-point mapping. Second, multiply this mapped curve by the following exponential function to correct the area problem:

EQUATION 34-1
Correction needed when converting a pdf from the linear to the base ten logarithmic number line.

$$c(g) = \ln(10)\, 10^{g}$$

There is also another way to look at this issue. A histogram is created for a group of number by breaking the linear number line into equally spaced bins. But how would this histogram be created on the logarithmic scale? There are two choices. First, you could calculate the histogram on the linear scale, and then transfer the value of the bins to the log scale. However, the equally spaced bins on the linear scale become unequally spaced on the log scale, and Eq. 34-1 would be needed as a correction. Second, you could break the logarithmic number line in equally spaced bins, and directly fill up these bins with the data. This procedure accurately estimates the pdf on the log scale without any additional corrections.

Now back to Fig. 34-4a. The example shown is a Gaussian (normal) curve with a mean of -0.25 and a standard deviation of 0.25, measured on the base ten logarithmic number line. Since it is a normal distribution when displayed on the logarithmic scale, it is given the special name: **log-normal**. When this pdf is displayed on the linear scale it looks entirely different, as we will see shortly. About 95% of the numbers generated from a normal distribution lie within +/- 2 standard deviations of the mean, or in this example, from -0.75 to 0.250, on the log scale. Converting back to the linear scale, this particular random process will generate 95% of its samples between $10^{-0.75}$ and $10^{0.25}$, that is, between 0.178 and 1.778.

The important point is that this is a single process that generates numbers, but we can look at those numbers on either the linear or the logarithmic scale. For instance, on the linear scale the numbers might look like: 1.2034, 0.3456, 0.9643, 1.8567, and so on. On the log scale these same numbers would be log(1.2034) = 0.0804, -0.4614, -0.0158, 0.2687, respectively. When we ask if this distribution follows Benford's law, we are referring to the numbers on the linear scale. That is, we are looking at the leading digits of 1.2034, 0.3456, 0.9643, 1.8567, etc. However, to understand why Benford's law is being followed or not followed, we will find it necessary to work with their logarithmic counterparts.

The next step is to determine what fraction of samples produced by this pdf have 1 as their leading digit. On the linear number line there are only certain regions where a leading digit of 1 is produced, such as: 0.1 to 0.199999; 1 to 1.99999; 10 to 19.9999; and so on. The corresponding locations on the base ten log scale are: -1.000 to -0.699; 0.000 to 0.301; and 1.000 to 1.301, respectively. In Fig. 34-4b these regions have been marked with a value of one, while all other sections of the logarithmic number line are given a value of zero. This allows the waveform in Fig. (b) to be used as a **sampling function**, and therefore we will call it, **sf(g)**.

Here is how it works. We multiply *pdf(g)* by *sf(g)* and display the result in Fig. (c). As shown, this isolates those sections of the pdf where 1 is the leading digit. We then find the total area of these regions by integrating from negative to positive infinity. Now you can see one

reason this analysis is carried out on the logarithmic number line: *the sampling function is a simple periodic pattern of pulses*. In comparison, think about how this sampling function would appear on the linear scale– far too complicated to even consider.

The above procedure is expressed by the equation in (d), which calculates the fraction of number produced by the distribution with 1 as the leading digit. However, as before, even if this number is exactly 0.301, it would not be conclusive proof that the pdf follows Benford's law. To show this we must conduct the Ones Scaling Test. That is, we will adjust *pdf(g)* such that the numbers it produces are multiplied by a constant that is slightly above unity. We then recalculate the fraction of ones in the leading digit position, and repeat the process many times.

Here we find a second reason to use the logarithmic scale: *multiplication on the linear number line becomes addition in the logarithmic domain*. On the linear scale we calculate: $n \times 1.01$, while on the logarithmic scale this becomes: $log(n) + log(1.01)$. In other words, on the logarithmic number line we scale the distribution by adding a small constant to each number that is produced. This has the effect of shifting the entire *pdf(g)* curve to the right a small distance, which we represent by the variable, ***s***. This is shown in Fig. (f). Mathematically, shifting the signal *pdf(g)* to the right a distance, *s*, is written *pdf(g-s)*.

The sampling function in Fig. (g) is the same as before; however, it now isolates a different section of the pdf, shown in (h). The integration also goes on as before, with the addition of the shift, *s*, represented in the equation. In short, we have derived an equation that provides the probability that a number produced by *pdf(g)* will have 1 in the leading digit position, for any scaling factor, *s*. As before, we will call this the Ones Scaling Test, and denote it by **ost(s).** This equation is given in (i), and reprinted below:

EQUATION 34-2
Calculating the Ones Scaling Test from the probability density function, by use of a scaling function. This equation also appears in Fig. 34i.

$$ost(s) = \int_{-\infty}^{+\infty} sf(g)\ pdf(g-s)\ dg$$

The signal *ost(s)* is nothing more than a continuous version of the graphs shown in Fig. 34-3. If *pdf(g)* follows Benford's law, then *ost(s)* will be approximately a constant value of 0.301. If *ost(s)* deviates from this key value, Benford's law is not being followed. For instance, we can easily see from Fig. (e) that the example pdf in (a) does not follow the law.

These last steps and Eq. 34-2 should look very familiar: *shift, multiply, integrate*. That's convolution! Comparing Eq. 34-2 with the definition

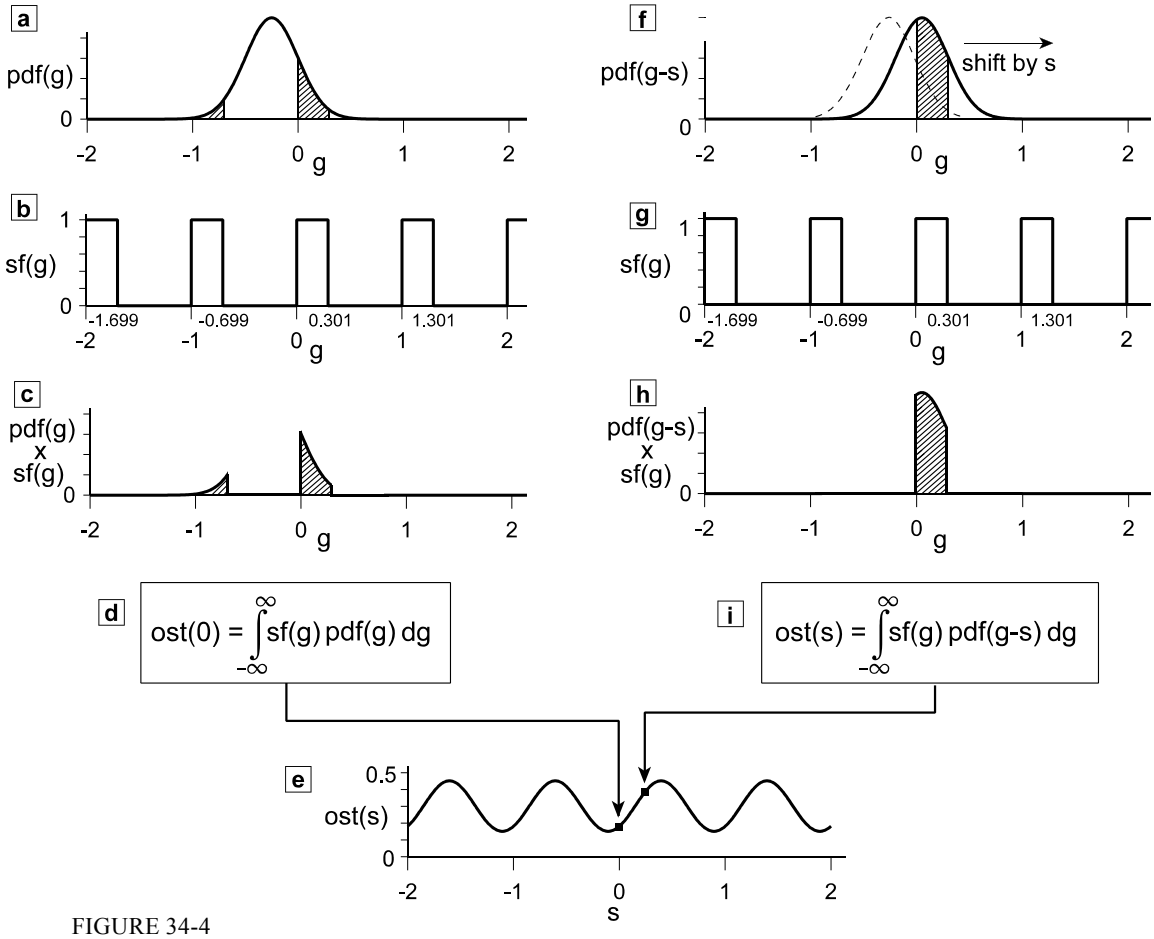


FIGURE 34-4
Expressing Benford's law as a convolution. Figures a-e show how to calculate the probability that a sample produced by *pdf(g)* will have a leading digit of 1. Figures f-i extend this calculation into the complete Ones Scaling Test. This shows that the Ones Scaling Test, *ost(g)*, is equal to the convolution of the probability density function, *pdf(g)*, and the scaling function, *sf(g)*.

of convolution (Eq. 13-1 in chapter 13), we have succeeded in expressing Benford's law as a straightforward linear system:

EQUATION 34-3
Benford's law written as a convolution. The negative sign in *pdf(-g)* is an artifact of how the equation is derived and is not important

$$ost(g) = sf(g) * pdf(-g)$$

There are two small issues that need to be mentioned in this equation. First, the negative sign in *pdf(-g)*. As you recall, convolution requires that one of the two original signals be flipped left-or-right before the shift, multiply, integrate operations. This is needed for convolution to properly represent linear system theory. On the other hand, this flip not needed in examining Benford's law; it's just a nuisance. Nevertheless, we need to account for it somewhere. In Eq. 34-3 we account for it by

pre-flipping *pdf(g)* by making it *pdf(-g)*. This pre-flip cancels the flip inherent in convolution, keeping the math straight. However, the whole issue of using *pdf(-g)* instead of *pdf(g)* is unimportant for Benford's law; it disappears completely in the next step.

The second small issue is a signal processing notation, the elimination of the variable, ***s***. In Fig. 3-4 we write *pdf(g)* and *sf(g)*, meaning that these two signals have the logarithmic number line as their independent variable, *g*. However, the Ones Scaling Test is written *ost(s)*, where *s* is a *shift* along the logarithmic number line. This distinction between *g* and *s* is needed in the derivation to understand how the three signals are related. However, when we get to the shorthand notation of Eq. 34-3, we eliminate *s* by changing *ost(s)* to *ost(g)*. This places the three signals, *pdf(g)*, *sf(g)* and *ost(g)* all on equal footing, each running along the logarithmic number line.

Solving in the Frequency Domain

Figure 34-5 is what we have been working toward, a systematic way of understanding the operation of Benford's law. The left three signals, the **logarithmic domain**, are *pdf(g)*, *sf(g)* and *ost(g)*. The particular examples in this figure are the same ones we used previously (i.e., Fig. 34-4). These three signals are related by convolution (Eq. 34-3), a mathematical operation that is not especially easy to deal with. To overcome this we move the problem into the **frequency domain** by taking the Fourier transform of each signal. Using standard DSP notation, we will represent the Fourier transforms of *pdf(g)*, *sf(g)*, and *ost(g)*, as *PDF(f)*, *SF(f)*, and *OST(f)*, respectively. These are shown on the right side of Fig. 34-5.

By moving the problem into the frequency domain we replace the difficult operation of convolution with the simple operation of multiplication. That is, the six signals in Fig. 34-5 are related as follows:

$$\begin{array}{ccccc} ost(g) & = & sf(g) & * & pdf(-g) \\ \updownarrow FT & & \updownarrow FT & & \updownarrow FT \\ OST(f) & = & SF(f) & \times & PDF^{*}(f) \end{array}$$

EQUATION 34-4
The Fourier transform converts the difficult operation of convolution into a simple multiplication.

A small detail: The Fourier transform of *pdf(g)* is *PDF(f)*, while the Fourier transform of *pdf(-g)* is *PDF*(f)*. The star in *PDF*(f)* means it is the **complex conjugate** of *PDF(f)*, indicating that all of the phase values are changed in sign. However, notice that Fig. 34-5 only shows the magnitudes; we are completely ignoring the phases. The reason for this is simple– the phase does not contain information we are interested in for this particular problem. This makes it unimportant if we use *pdf(g)* vs. *pdf(-g)*, or *PDF(f)* vs. *PDF*(f)*.

Notice how these signals represent the key components of Benford's law. First, there is a group of numbers or a probability density function that can generate a group of numbers. This is represented by *pdf(g)* and *PDF(f)*. Second, we modify each number in this group or distribution by taking its leading digit. This action is represented by convolving *pdf(g)* with *sf(g)*, or by multiplying *PDF(f)* by *SF(f)*. Third, we observe that the leading digits often have an unusual property. This unusual characteristic is seen in *ost(g)* and *OST(f)*.

All six of these signals have specific characteristics that are fixed by the definition of the problem. For instance, the value at f=0 in the frequency domain always corresponds to the average value of the signal in the logarithmic domain. In particular, this means that *PDF(0)* will always be equal to one, since the area under *pdf(g)* is unity. In this example we are using a Gaussian curve for *pdf(g)*. One of the interesting properties of the Gaussian is that its Fourier Transform is also a Gaussian, one-sided in this case, as shown in Fig. (d). These are related by $\sigma_f = 1/(2\pi\sigma_g)$.

Since *sf(g)* is periodic with a period of one, *SF(f)* consists of a series of spikes at f = 0, 1, 2, 3, ..., with all other values being zero. This is a standard transform pair, given by Fig. 13-10 in chapter 13. The zeroth spike, *SF(0),* is the average value of *sf(g)*. This is equal to the fraction of the time that the signal is in the high state, or *log*(2) - *log*(1) = 0.301. The remaining spikes have amplitudes: *SF(1)* = 0.516, *SF(2)* = 0.302, *SF(3)* = 0.064, and so on, as calculated from the above reference.

Lastly we come to *ost(g)* and *OST(f)*. If Benford's law is being followed, *ost(g)* will be a flat line with a value of 0.301. This corresponds to *OST(0)* = 0.301, with all other values in *OST(f)* being zero. However, if Benford's law is not being followed, then *ost(g)* will be periodic with a period of one, as show in Fig. (c). Therefore, *OST(f)* will be a series of spikes at f = 0, 1, 2, 3, ..., with the space between being zero.

Solving Mystery #1

There are two main mysteries in Benford's law. The first is this: ***Where does the logarithmic pattern of leading digits come from? Is it some hidden property of Nature?*** We know that *ost(g)* is a constant value of 0.301 if Benford's law is being followed. Using Fig. 34-5 we can find where this number originates. By definition, the average value of *ost(g)* is *OST(0);* likewise, the average value of *sf(g)* is *SF(0)*. However, *OST(0)* is always equal to *SF(0)*, since *PDF(0)* has a constant value of one. That is, the average value of *ost(g)* is equal to the average value of *sf(g)*, and does not depend on the characteristics of *pdf(g)*. As shown above, the average value of *sf(g)* is *log*(2) - *log*(1) = 0.301, which dictates that the average value of *ost(g)* is also 0.301. If we repeated this procedure looking for 2 as the leading digit, the average value of *sf(g)* would be *log*(3) - *log*(2) = 0.176. The remaining digits, 3-9, are handled in the same way. In answer to our question, the logarithmic pattern of leading digits derives

solely from *sf(g)* and the convolution, and not at all from *pdf(g)*. **In short, the logarithmic pattern of leading digits comes from the manipulation of the data, and has nothing to do with patterns in the numbers being investigated.**

This result can be understood in a simple way, showing how Benford's law resembles a magician's slight of hand. Say you tabulate a list of numbers appearing in a newspaper. You tally the histogram of leading digits and find that they follow the logarithmic pattern. You then wonder how this pattern could be hidden in the numbers. The key to this is realizing that something has been concealed— a big something.

Recall the program in Table 34-1, where lines 400-430 extract the leading digit of each number. This is done by multiplying or dividing each number repeatedly by a factor of ten until it is between 1 and 9.999999. This manipulation of the data is far from trivial or benign. You don't notice this procedure when manually tabulating the numbers because your brain is so efficient. But look at what this manipulation involves. For example, successive numbers might be multiplied by: 0.01, 100, 0.1, 1, 10, 1000, 0.001, and so on.

This changes the numbers in a pattern based on powers of ten, i.e., the *anti-logarithm*. You then examine the processed data and marvel that it looks *logarithmic*. Not realizing that your brain has secretly manipulated the data, you attribute this logarithmic pattern to some hidden feature of the original numbers. *Voila! The mystery of Benford's law!*

Solving Mystery #2

The second mystery is: ***Why does one set of numbers follow Benford's law, while another set of numbers does not?*** Again we can answer this question by examining Fig. 34-5. Our goal is to find the characteristics of *pdf(g)* that result in *ost(g)* having a constant value of 0.301. As shown above, the average value of *ost(g)* will always be 0.301, regardless if Benford's law is being followed or not. So our only concern is whether *ost(g)* has oscillations, or is a flat line.

For *ost(g)* to be a flat line it must have no sinusoidal components. In the frequency domain this means that *OST(f)* must be equal to zero at all frequencies above f=0. However, *OST(f)* is equal to $SF(f) \times PSF(f)$, and *SF(f)* is nonzero only at the integer frequencies, f = 0, 1, 2, 3, 4, and so on. Therefore, *ost(g)* will be flat, if and only if, *PSF(f)* has a value of zero at the integer frequencies. The particular example in Fig. 34-5 clearly does not meet this condition, and therefore does not follow Benford's law. In Fig. (d), *PDF(1)* has a value of 0.349. Multiplying this by the value of $SF(1) = 0.516$, we find $OST(1) = 0.18$. Therefore, *ost(g)* has a sinusoidal component with a period of one, and an amplitude of 0.18. This is a key result, describing what criterion a distribution must

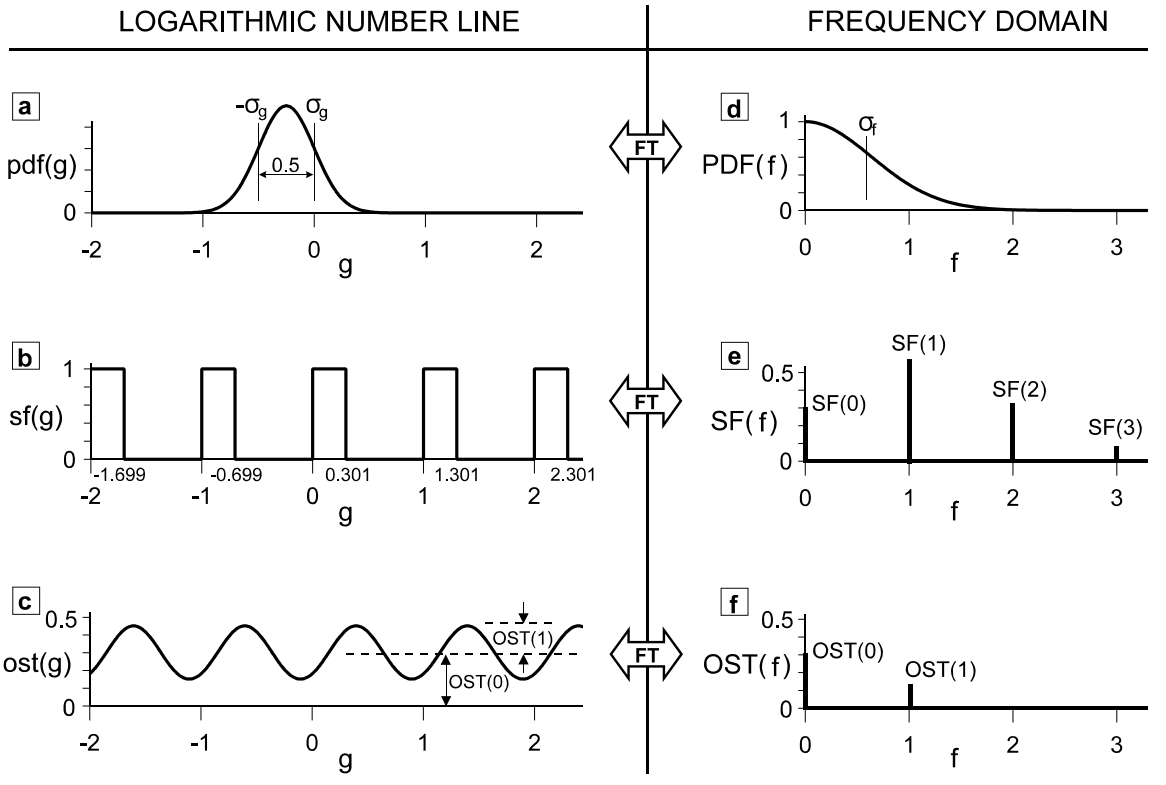


FIGURE 34-5
Benford's law analyzed in the frequency domain. In the logarithmic domain Benford's law is represented as a convolution, *ost(g) = sf(g) ✳ pdf(-g)*. In the frequency domain this becomes the much simpler operation of multiplication, *OST(f) = SF(f) × PDF*(f)*.

meet to follow Benford's law. This is important enough that we will express it as a theorem.

Benford's Law Compliance Theorem

Let P be a random process generating numbers in base B on the linear number line, pdf(g) its probability density function expressed on the base B logarithmic number line, and PDF(f) the Fourier transform of pdf(g). The numbers generated by P will follow Benford's law, if and only if, PDF(f) = 0 at all nonzero integer frequencies.

Our next step is to examine what type of distributions comply with this theorem. There are two distinct ways that *PDF(f)* can have a value of zero at the nonzero integer frequencies. As shown in Fig. 34-6b, *PDF(f)* can be oscillatory, periodically hitting zero at frequencies that include the integers. In the logarithmic domain this corresponds to two or more discontinuities spaced an integer distance apart, such as sharp edges or abrupt changes in the slope. Figure (a) shows an example of this, a rectangular pulse with edges at -1 and 1. These discontinuities can easily

be created by human manipulation, but seldom occur in natural or unforced processes. This type of distribution does follow Benford's law, but it is mainly just a footnote, not the bulk of the mystery.

Figure (d) shows a far more important situation, where *PDF(f)* smoothly decreases in value with increasing frequency. This behavior is more than common, it is the rule. It is what you would find for most any set of random numbers you examine. The key parameter we want to examine is how fast the curve drops to zero. For instance, the curve in Fig. 34-6d drops so rapidly that it has a negligible value at f=1 and all higher frequencies. Therefore, this distribution will follow Benford's law to a very high degree. Now compare this with Fig. 34-5d, an example where *PDF(f)* drops much slower. Since it has a significant value at f=1, this distribution follows Benford's law very poorly.

Now look at *pdf(g)* for the above two examples, Figs. 34-6c and 34-5a. Both of these are normal distributions on the logarithmic scale; the only difference between them is their width. A key property of the Fourier transform is the compression/expansion between the domains. If you need to refresh your memory, look at Figure 10-12 in chapter 10. In short, if the signal in one domain is made narrower, the signal in the other domain will become wider, and vice versa. For example, in Fig. 34-5a the standard deviation of *pdf(g)* is $\sigma_g = 0.25$. This results in *PDF(f)* having a standard deviation of: $\sigma_f = 1/(2\pi\sigma_g) = 0.637$. In Fig. 34-6 the log domain is twice as wide, $\sigma_g = 0.50$, making the frequency domain twice as narrow, $\sigma_f = 0.318$. In these figures the width of the distribution is indicated as 2σ, that is, $-\sigma$ to σ. This is common, but certainly not the only way to measure the width.

In short, if *pdf(g)* is narrow, then *PDF(f)* will be wide. This results in *PDF(f)* having a significant amplitude at f=1, and possibly at higher frequencies. Therefore, the distribution will not follow Benford's law. However, if *pdf(g)* is wide, then *PDF(f)* will be narrow. This results in *PDF(f)* falling near zero before f=1, and Benford's law is followed.

A key issue is how wide or narrow *pdf(g)* needs to be to toggle between the two behaviors. To follow Benford law, *PDF(f)* must drop to near zero by f=1. Further, f=1 in the frequency domain corresponds to a sinusoid with a period of *one* on the log scale, making this the critical distance. This gives us the answer to our question. **With a few caveats, Benford's law is followed by distributions that are wide compared with unit distance along the logarithmic scale. Likewise, the law is not followed by distributions that are narrow compared with unit distance.**

To be clear, one exception occurs when *PDF(f)* is oscillatory such as in Fig. 34-6b. The other exception is when *PDF(f)* does not smoothly decreases in value with increasing frequency. Also, the definition of "width" used here is slightly fuzzy. We will improve upon this in the next section. However, these are minor issues and details; do not let them distract from your understanding of the mainstream phenomenon.

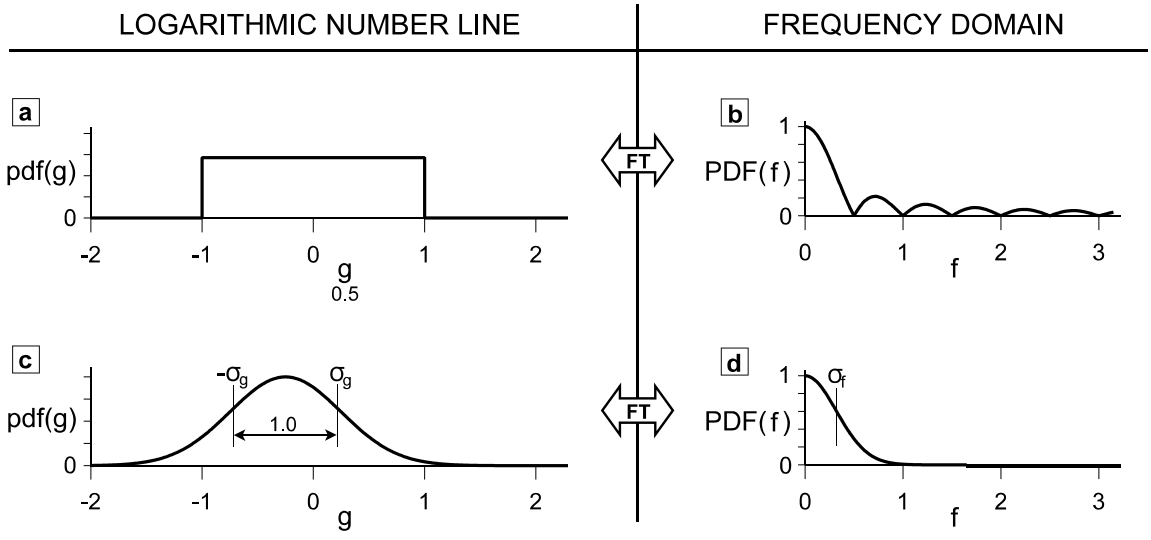


FIGURE 34-6
Two ways of obeying Benford's law. The Benford's Law Compliance Theorem shows that a distribution will obey the law only if *PDF(f)* has a value of zero at f = 1, 2, 3, ... This can be achieved in two different ways. In (b) the oscillations hit zero at these frequencies, while in (d) the curve has dropped to zero before f=1.

More on Following Benford's law

This last result is very surprising; the mystery of Benford's law turns out to be nothing more than distribution width. Figure 34-7 demonstrates this using our previous examples. Figures (a) and (c) are the histograms of the income tax return and the RNG numbers, respectively, on the logarithmic scale. Figure (b) and (d) are their Fourier Transforms. The Benford's Law Compliance Theorem tells us that (b) will follow Benford's law very closely, while (d) will follow it very poorly. That is, *PDF(f)* falls to near zero before f=1 for the income tax numbers, but does not for the RNG numbers. The next step of this is less rigorous, but still perfectly clear. Figure (b) falls to zero quickly because (a) is broad. Likewise, (d) falls to zero more slowly because (c) is narrow.

This also tells us something about the magic trick. If the distribution is wide compared with unit distance on the log axis, it means that the spread in the set of numbers being examined is much greater than *ten*. For instance, look back at the income tax numbers shown in Fig. 34-2a. The largest numbers in this set are about a *million* times greater in value than the smallest numbers. This extensive spread is a key part of stamping the logarithmic pattern into the data. That is, 543,923,100 must be divided by *100,000,000* to place it between 1 and 9.99999, while 1,221 only needs to be divided by *1,000*. In other words, different numbers are being treated differently, all according to an anti-logarithmic pattern.

Now look at the RNG numbers in Fig. 34-2, a group that does not obey Benford's law. The largest numbers in this set are about four times the smallest numbers (measured from -σ to +σ). That is, they are grouped relatively close together in value. When we extract the leading digits from these numbers, most of them are treated exactly the same. For instance, both 7.844026 and 1.230605 are divided by 1 to place them between 1 and 9.999999. Likewise, numbers clustered around 5,000 would all be divided by 1,000 to extract the leading digits. Since the vast majority of the numbers are being treated the same, or nearly the same, the distortion of the data is relatively weak. That is, the logarithmic pattern cannot be introduced into the data, and the magic trick fails.

How does Benford's law behave in other bases? Suppose you repeat the previous derivation in base 4 instead of base 10. The base 4 logarithmic number line is used and the Benford's Law Compliance Theorem still holds. The difference comes in when we compare the width of our test distribution with one unit of distance on the logarithmic scale. One unit of distance in base 4 is only $\log_{10}(4) = 0.602$ the length of one unit in base 10, making it easier for the distribution to comply with Benford's law. In terms of the magic trick, the spread in the numbers being examined only needs to be much greater than *four*, rather than *ten*. In the common case where *PDF(f)* smoothly decreases, Benford's law will always be followed better when converted to a lower base, and worse if converted to a higher base. For instance, the income tax numbers *will not* follow Benford's law if converted to base 10,000 or above (making the unit distance on the log scale four times greater). Likewise, the RND number *will* follow Benford's law if converted to base 2 (shortening the unit distance to $\log_{10}(2) = 0.301$).

A note for advanced readers: You may have noticed a problem with this last statement, that is: *all numbers in base 2 have a leading digit of 1*. However, a more sophisticated definition of Benford's law can be used to eliminate issues of this sort. The leading digit of a number can be found by repeatedly multiplying/dividing the number by ten until it is between 1 and 9.99999, and then taking the integer portion. The advanced method stops after the first step, and directly looks at the pdf of the numbers running between 1 and 9.99999. We will call these the **modified numbers**. If Benford's law is being followed, $a(n) = k/n$, where $a(n)$ is the probability density function of the modified numbers on the linear scale, and k is a constant providing unity area under the pdf curve. If needed for some purpose, we can find the fraction of numbers that have a leading digit of 1 by integrating $a(n)$ from 1 to 2. Since the integral of k/n is the logarithm, if Benford's law is being followed this fraction is given by: $log(2) - log(1) = 0.301$. That is, we can easily move from the advanced representation to the simpler leading-digit definition.

This "k/n" form of Benford's law can be also derived from the method of Fig. 34-5. The fraction of the modified numbers that are greater than p but less than q is found by integrating $a(n)$ between p and q. Further, this fraction will remain a constant under the scaling test if Benford's law is

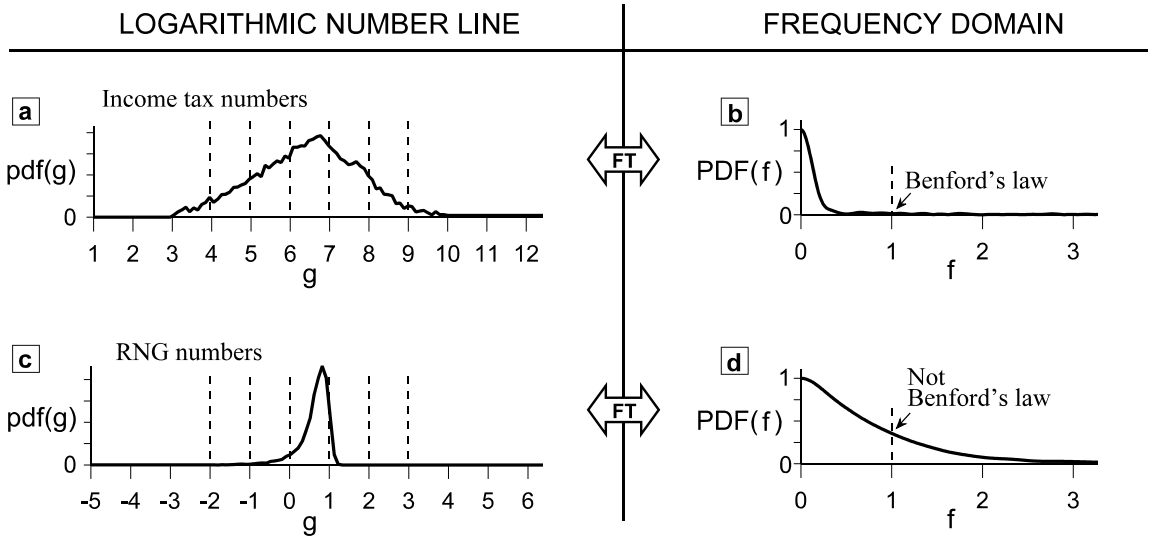


FIGURE 34-7
Two examples for understanding Benford's law. A distribution will follow Benford's law only if *PDF(f)* falls to near zero before f=1 (excluding the oscillatory case). In turn, this requires that *pdf(g)* be broad compared with one unit of distance on the logarithmic scale. This explains why the income tax numbers follow the law, while the RNG numbers do not.

being followed. However, this value is also equal to the average value of the appropriate scaling function. The logic here is the same used to show that the average value of *ost(g)* is equal to the average value of *sf(g)* in "Solving Mystery #1." These two factors become the left and right sides of the following equation, respectively:

EQUATION 34-5
Derivation of *k/n* form of Benford's law.

$$\int_{p}^{q} a(n)\,dn = \log(p) - \log(q)$$

Solving this equation results in Benford's law, i.e., $a(n) = k/n$,

Analysis of the Log-Normal Distribution

We have looked at two log-normal distributions, one having a standard deviation of 0.25 and the other a standard deviation of 0.5. Surprisingly, one follows Benford's law extremely well, while the other does not follow it at all. In this section we will examine the analytical transition between these two behaviors for this particular distribution.

As shown in Fig. 34-5d, we can use the value of *OST(1)* as a measure of how well Benford's law is followed. Our goal is to derive an equation relating the standard deviation of *psf(g)* with the value of *OST(1)*, that is, relating the width of the distribution with its compliance with Benford's law. Notice that this has rigorously defined the problem (removed the

fuzziness) by specifying three things, the shape of the distribution, how we are measuring compliance with Benford's law, and how we are defining the distribution width.

The next step is to write the equation for *PSF(f)*, a one-sided Gaussian curve, having a value of zero at f=0, and a standard deviation of σ_f:

$$PSF(f) = e^{-f^2/2\sigma_f^2}$$

Next we plug in the conversion from the logarithmic-domain standard deviation, $\sigma_f = 1/(2\pi\sigma_g)$, and evaluate the expression at f=1:

$$PSF(1) = e^{-2\pi^2\sigma_f^2}$$

Lastly, we use $OST(1) = SF(1) \times PSF(1)$, where $SF(1) = 0.516$, to reach the final equation:

EQUATION 34-5
Compliance of the log-normal distribution with Benford's law.

$$OST(1) = 51.6\% \times e^{-2\pi^2\sigma_f^2}$$

As illustrated in Fig. 34-5c, the highest value in *ost(g)* is *OST(1)* plus 0.301, and the lowest value is 0.301 - *OST(1)*. These highest and lowest values are graphed in Fig. 34-8a. As shown, when the 2σ width of the distribution is 0.5 (as in Fig 34-5a), the Ones Scaling Test will have values as high as 45% and as low as 16%, a very poor match to Benford's law. However, doubling the width to $2\sigma = 1.0$ results in a high to low fluctuation of less than 1%, a good match.

There are a number of interesting details in this example. First, notice how rapidly the transition occurs between following and not following Benford's law. For instance, two cases are indicated by A and B in Fig. 34-8, with $2\sigma = 0.60$ and $2\sigma = 0.90$, respectively. In Fig. (b) these are shown on the linear scale. Now imagine that you are a researcher trying to understand Benford's law, before reading this chapter. Even though these two distributions appear very similar, one follows Benford's law very well, and the other doesn't follow it at all! This gives you an idea of the frustration Benford's law has produced.

Second, even though the curves in Fig. (a) move together extremely rapidly, they never actually meet (excluding infinity which isn't allowed for a pdf). For instance, from Eq. 34-5 a log-normal distribution with a standard deviation of three will follow Benford's law within about 1 part in 100,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000. That's pretty close! In fact, you could not statistically detect this error even with a billion computers, each generating a billion numbers each second, since the beginning of the universe.

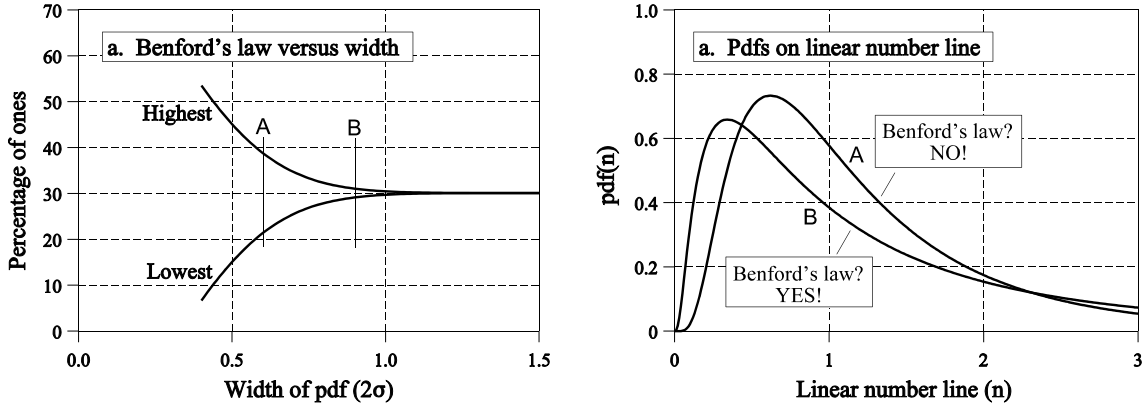


FIGURE 34-8
Analyzing the log-normal distribution for complying with Benford's law. Even a slight difference in the width of this distribution, shown by A and B, can drastically change its following the law.

Nevertheless, this is a finite error, and has caused frustration of its own. Again imagine that you are a researcher trying to understand Benford's law. You proceed by writing down some equation describing when Benford's law will be followed, and then you solve it. The answer you find is— *Never!* There is no distribution (excluding the oscillatory case of Fig. 34-6b) that follows Benford's law exactly. An equation doesn't give you what is *close*, only what is *equal*. In other words, you find no understanding, just more mystery.

Lastly, the log-normal distribution is more than just an example, it is an important case where Benford's law arises in Nature. The reason for this is one of the most powerful driving forces in statistics, the *Central Limit Theorem* (CLT). As discussed in chapter 2, the CLT describes that *adding* many random numbers produces a normal distribution. This accounts for the normal distribution being so commonly observed in science and engineering. However, if a group of random numbers are *multiplied*, the result will be a normal distribution on the logarithmic scale. Accordingly, the log-normal distribution is also commonly found in Nature. This is probably the single most important reason that some distributions are found to follow Benford's law while others do not. Normal distributions are not wide enough to follow the law. On the other hand, broad log-normal distributions follow it to a very high degree.

Want to generate numbers that follow Benford's law for your own experiments? You can take advantage of the CLT. Most computer languages have a random number generator that produces values uniformly distributed between 0 and 1. Call this function multiple times and multiply the numbers. It can be shown that $PDF(1) = 0.344$ for the uniform distribution, and therefore the product of these numbers follows Benford's law according to $OST(1) = 51.6\% \times 0.344^{\alpha}$, where α is how many random numbers are multiplied. For instance, ten multiplications produce a random number that comes from a log-normal distribution with

a standard deviation of approximately 0.75. This corresponds to *OST(1)* = 0.0012%, a very good fit to Benford's law.

If you do try some of these experiments, remember that the statistical variation (noise) on *N* random events is about SQRT(*N*). For instance, suppose you generate 1 million numbers in your computer and count how many have 1 as the leading digit. If Benford's law is being followed, this number will be about 301,000. However, when you repeat the experiment several times you find this changes randomly by about 1,000 numbers, since SQRT(1,000,000) = 1,000. In other words, using 1 million numbers allows you to conclude that the percentage of numbers with one as the leading digit is about 30.1% +/- 0.1%. As another example, the ripple in Fig. 34-3a is a result of using 14,414 samples. For a more precise measurement you need more numbers, and it grows very quickly. For instance, to detect the error of *OST(1)* = 0.0012% (the above example), you will need in excess of a *billion* numbers.

The Power of Signal Processing

Benford's law has never been viewed as a major mathematical problem, only a minor mystery. Nevertheless, many bright and creative people have spent time trying to understand it. The primary goal of this chapter has been to demonstrate the power of DSP in nontraditional applications. In the case of Benford's law this power is clear; signal processing has succeeded where other mathematical techniques have failed.

Nowhere is this more apparent than a review article published in 1976 by mathematician Ralph Raimi. He examined the many approaches in explicit mathematical detail, and his paper has become a landmark in the history of this problem. Buried in the detailed math, Raimi makes the brief comment: *"...many writers ... have said vaguely that Benford's law holds better when the distribution ... covers several orders of magnitude*." As we now know, this is the root of the phenomenon. In one of the most colorful events of this history, a small error in logic prompted Raimi to argue that this could not be correct. [Specifically, scaling a distribution *does not* change how many orders of magnitude it covers.] While this slight misdirection probably made no difference, it shows just how little success had been achieved by traditional mathematics. An understanding of the basic operation of Benford's law was nowhere on the horizon.

Lastly, this discussion would be incomplete without mentioning the practical applications of Benford's law. Next time you file your income tax return or other financial report, consider what happens to the distribution of leading digits if you fabricate some of the numbers. I'm not going to help you cheat, so I won't give the details away. Simply put, the numbers you make up will probably not follow Benford's law, making your fraudulent report distinguishable from an accurate one. I'll let you imagine who might be interested in this.