A Benford primer

Rik King

Abstract

This paper aims to excite the curiosity of the reader to achieve a basic level of understanding of the meaning of Benford’s law. Benford’s law concerns the prevalence of first and subsequent digits which appear in naturally occurring numerical transactions. One possible forensic application is in the detection of fraud in machine-generated sets of data, which do not obey this law. In addition to explanation with illustrative examples, some programs constructed in Excel and R are provided.

Keywords: Benford, Excel, first digit, fraud, primer, R, scale invariance, second digit.

Introduction

It is thought that Mathematics dates from the Sumerian civilization circa 3000 BC., from which many centuries of mathematical research displaying sophistication and complexity follow. So there is something surprising and very wonderful about the simplicity of a discovery depending merely on the numbers 0,…,9. This has popped up in the modern era; and with no insignificant mathematical curiosity, it turns out to be the focus of a very intense, expanding current research area.

This short article aims to excite the curiosity of the reader to achieve a basic level of understanding of the meaning of Benford’s law. In addition to explanation with examples, some programs constructed in Excel and R are meant to be worked through in a ‘learn as you go’ style.

The leading digit

To begin with, think about the numbers which form so many of our transactions in everyday life. Usually, these are greater than 0, that is, positive. Now, each number has what is called a leading digit, which is just the first part of the number on its own. So, for example, the leading digits of the numbers 7.62, 34 and 0.0528 are 7, 3 and 5 respectively. A leading digit can be any one of 1, . . . ,9 (0 is not in the list because then the number would be 0, and not of any interest). Presented with a large file of numbers to be examined, intuition would suggest all the leading digits i.e 1 to 9, should appear as frequently as each other - as many ones as twos as threes etc. as nines - indeed, why should it be otherwise? In Statistics, however, intuition often misleads, which is how it turns out in this case.

The surprising fact is, that, in very many collections of numbers, there are fixed proportions of leading digits 1 to 9, and these proportions are far from equal. The values are given by Benford’s Law (Benford, F.,1938), a statement of which is as follows:

The proportions for d being the first digit of a number are given approximately by

$$\log_{10}(1 + \frac{1}{1+d}) \qquad (1)$$

where d is any of (1,…,9). This gives rise to Table 1 below:

Table 1: Benford's Law

Digit	1	2	3	4	5	6	7	8	9
Proportion	0.301	0.176	0.125	0.097	0.079	0.067	0.058	0.051	0.046

So, this means that in a collection of numbers there should be about 30% ones, 17% twos - just under 50% ones and twos combined, but only 5% nines. Figure 1 displays these proportions.

Thus, it turns out that the first digits of numbers in many collections of data, rather than following a uniform distribution (equal amounts), follow a discrete logarithmic distribution, unusual, because it is not described with the help of any parameter: the formula involves only d, the digit number, and no other variable. The logarithmic term may be to any base, but base 10 is most frequently used.

Frank Benford was a physicist and the connection between his work and that of an earlier (1881) researcher Simon Newcomb, who was an astronomer, is truly fascinating. Stoessiger (2013) gives an interesting account of the link between the work of the two researchers.

Exploring a data file

The working for this section follows the method of Lanham (2019), which details how to obtain a large data source suitable for investigation. For every state in the USA, there is a State Occupational Employment and Wage Estimates (SOEWE) document, which provides information on employment and wage estimates for various occupations, with data collected directly from employers in all industry sectors. Data sets for different years are downloadable from the US Bureau of Labor Statistics[1].

This section is best read by working step-by-step through a data file, as described below after downloading into Excel a data set for any year from SOEWE. Here we discuss the SOEWE (2019) data set comprised originally of 36,383 records, reduced to 34,853 entries after sorting and the removal of incomplete lines.

Some simple Excel commands have been used to extract the first digit for each of the 34,853 records, and then collect the digits into groups and count the number in each group. The cleaned example file SOEWE.xls displays the following columns:

(i) Column C: the employment numbers

[1] *https : //www.bls.gov/oes/current/oessrcst.htm*

(ii) Column D, the first digit of each number in column C, extracted by the Excel function LEFT(.)

(iii) Columns F - N, the count of digits in column D, extracted by the Excel function COUNTIF(.,.)

The final proportions for digits 1 to 9 do not appear on the spreadsheet but should be calculated by the reader and checked from Tāble 2 and Figure 1 below:

Table 2: SOEWE result

Digit	1	2	3	4	5	6	7	8	9
Proportion	0.298	0.167	0.123	0.101	0.083	0.069	0.058	0.054	0.044

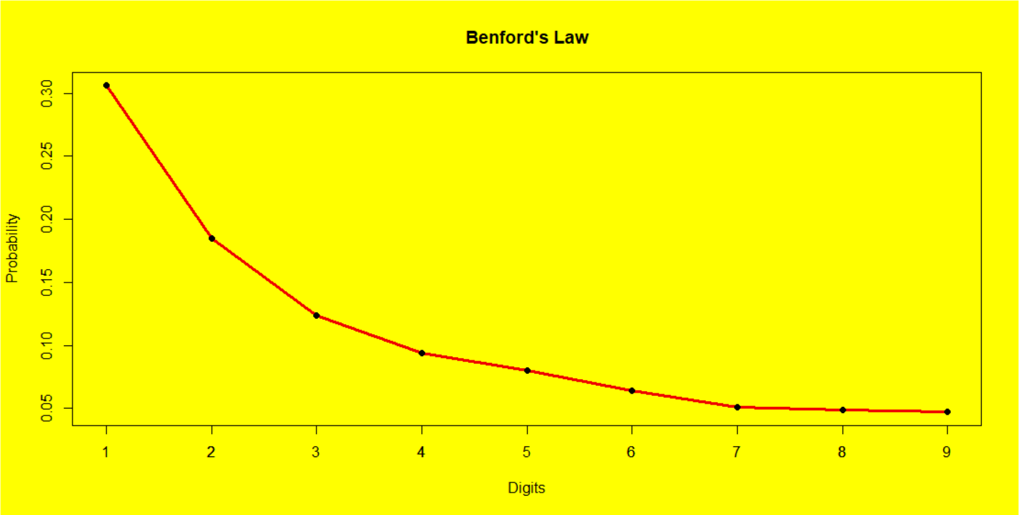


Figure 1: Benford’s Law.

In Figure 2 results are compared with the probabilities noted by Benford, showing good agreement (helped along by the large size of the SOEWE file).

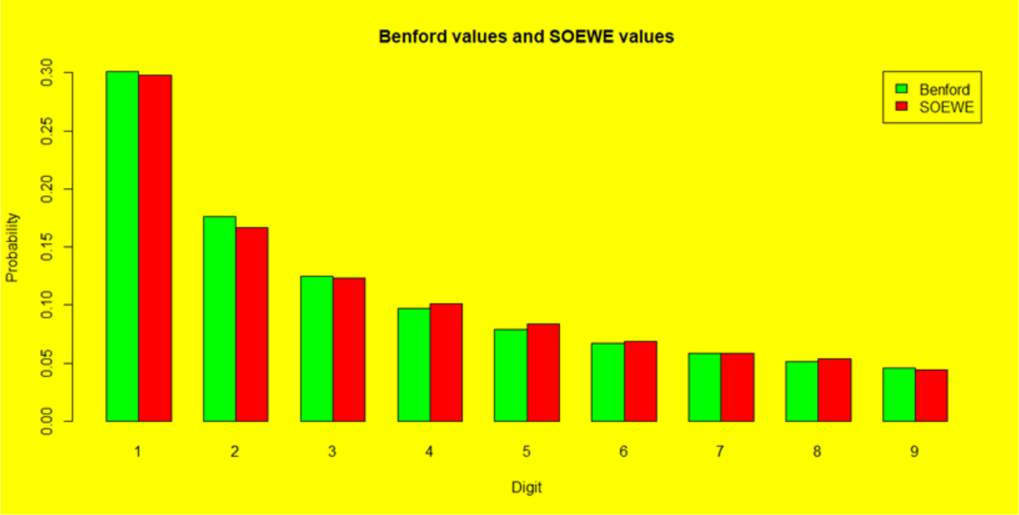


Figure 2: Benford values and SOEWE values.

Using R

While Excel spreadsheet is a useful method for processing large data files, the same results may be achieved through longer processes by using R. There is a sophisticated R package entitled BenfordAnalysis.R due to Cinelli (2018) which downloads data from several formats and conducts analysis at elementary and higher levels.

Fibonacci numbers

It is known that many mathematical sequences obey Benford's Law and amongst these is the well-known sequence due to Fibonacci where numbers are defined by the recurrence relation $F_{n+2} = F_{n+1} + F_n$, with $F_0 = 0$ and $F_1 = 1$. However, from the demonstration point of view, it is unfortunate that the terms grow large very quickly, and even using an efficient recursion, generating a large data file is beyond the capacity of an average laptop. A bypass of this difficulty is provided by Binet's approximation (Miller, 2015), which is given by:

$$F_n = \frac{1}{\sqrt{5}}\Big(\frac{1+\sqrt{5}}{2}\Big)^n - \frac{1}{\sqrt{5}}\Big(\frac{1-\sqrt{5}}{2}\Big)^n \qquad (2)$$

Since, however, terms with n large are to be generated, it is useful to simplify the above to:

$$F_n = \frac{1}{\sqrt{5}}\Big(\frac{1+\sqrt{5}}{2}\Big)^n \qquad (3)$$

This second form of Binet's approximation is employed in the program Fib.R, listed in Appendix (i). There, 1000 Fibonacci numbers are generated and the leading digit extracted; a count is made of the digits 1 to 9 and their proportions of the total (1000) calculated. The results, showing good agreement with the Benford values of Table 1, are displayed in Table 3.

Table 3: Benford's Law & Fibonacci numbers

Digit	1	2	3	4	5	6	7	8	9
Benford	0.301	0.176	0.125	0.097	0.079	0.067	0.058	0.051	0.046
Fibonacci	0.301	0.177	0.125	0.096	0.080	0.067	0.056	0.053	0.045

The Fibonacci numbers are discrete - i.e. in distinct units. It is useful now to look at a different growth process, a continuous set of numbers, this time drawn from Finance.

Finance

The material of this section follows the method of Miller (2015), which begins with the well-known formula for the amount \$A, accruing from an amount of \$P invested for n years at a rate r% per annum:

$$A = P(1 + 0.0r)^n \qquad (4)$$

An adaptation of the above formula is useful for considering the leading digits of the amounts. If d is the leading digit of an amount invested, after n years, the amount grows and the leading digit d moves up by 1 to become (d + 1). Thus, amount d(1 + 0.0r)^n = (d + 1). Solving for the number of years n is more appropriately done with logarithms to the base 10, rather than the usual natural logarithms, since the digits 1...9 are involved:

$$n = \frac{\log_{10}(\frac{d+1}{d})}{\log_{10}(1.0r)} \qquad (5)$$

From the foregoing format, it is possible to answer the following question: how long does it take for a deposit of $1 to grow to over $2, then how long for over $2 to become over $3 and so on? Notice that there will be intervals during which the leading digits of the amounts will be respectively 1, 2, 3,...,9. The quantities of interest are the lengths of the intervals during which the leading digit stays the same. This is because any count of digit frequencies will show a higher reading for a leading digit, which persists for a longer interval.

Listed in Appendix (ii) is the program My Deposit.R. It accepts the arguments amount and interest rate (no % sign) compounded annually, time being taken as indefinitely long. The function first needs to be defined in R, after which it is ready to be run for any specific example.

The following paragraph uses the results from that program, which the reader is encouraged to run. Consider the results from My Deposit (10, 5) which describes $10 which has been invested at 5% per annum.

The output appears in two sections. The first header viz. ”time to next digit up” indicates in years, how long it takes for the $10 to become $20, then $20 to become $30, then $30 to become $40, all the way up to $90: in other words, how long an amount remains with the same leading digit before jumping up to the next digit. So, reading from the output shows that it takes 14.2 years for $10 to become $20 and 8.3 years for $20 to become $30, etc. Therefore there were many more leading digit ones than twos, because of the relative lengths of the time intervals in which the leading digit did not jump up to the next value. The times for a digit to “jump up” steadily decrease. The second part of the output with the header time as a fraction of total time: 1 - 9 lists each time spent with a leading digit as a fraction of the overall time for the money to get from $10 to $90. For the example above, going from $10 to $20 took 0.301 of the total time, but from $80 to $90 only 0.045.

The program should be run several times for varying initial amounts when it becomes apparent that the second part of the output - a fraction of total time - is always the same. This is because it is the ratios of quantities (due to the interest rate) that count, and not the quantities themselves: it takes as long to get from $100 to $200 as from $10 to $20. Also, running the program for varying interest rates results always in the same fractional times. The core of the calculation - what is actually happening in the program - is exhibited in Table 4 below: The right-hand column, is, of course, just the Benford numbers.

Table 4: Program results

Digit	Log - Log	Result
1	log 2 - log 1	0.301
2	log 3 - log 2	0.176
3	log 4 - log 3	0.125
4	log 5 - log 4	0.097
5	log 6 - log 5	0.079
6	log 7 - log 6	0.067
7	log 8 - log 7	0.058
8	log 9 - log 8	0.051
9	log 10 - log 9	0.046

The above example is highly simplified - so, in particular, invested amounts are often not constant and interest rates typically vary over time, however, for the sake of simplicity in the demonstration, these complicating factors have been ignored.

There is a further very important point connected with My Deposit.R. It worked for dollar units of investment, but it could have worked just as well for an investment of the same amount converted to francs or yen - the final results would be connected by the currency conversion factors. The behaviour depended not on quantities, but on the ratios of quantities, governed by the interest rate: this is a point amplified in a later discussion on scaling.

Applications

Benford's law is not universal. The data must show a full order of values from 1 to 9 fold, and must not have imposed natural maxima or minima e.g. the petal sizes of a particular species of flower would not be suitable data. It does not apply to manufactured numbers such as car license plates, telephone numbers or bank account numbers. In general, it holds for natural numbers such as the area of landmasses, and volumes of river flow; and for many fundamental physical constants and quantities from the natural sciences.

A big area of application is finance (the topic of the final section). Various types of data certainly follow the law very accurately: these include stocks, shares, mathematical combination of numbers, such as quantity multiplied by price disbursements, and sales numbers.

A cautionary remark, however about general criteria for Benford suitability is in order. The great expansion in the diversity of applications, never envisaged even a few short years ago e.g. intensities in digital imaging, have consigned some previously held certainties into the hypothesis category, so now some writers e.g. Miller (2019) refer to previously held certainties on applicability as hypotheses, so e.g. the 'spread hypothesis'. There are now, however, formal approaches to proving whether a system satisfies Benford's Law. (cf. Berger & Hill, 2011).

Their work also provides the answer to the following question which may cross the mind of the reader: Is it possible to manufacture a set of numbers, which with certainty, will display the values of the Benford distribution? The short answer is ‘yes’.

To see that this is so, run the program Benford.R to be found in Appendix (iii). It simulates 5000000 numbers; a typical output is shown in Table 5 below:

Table 5: Benford’s Law & simulated numbers

Digit	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Benford	0.301	0.176	0.125	0.097	0.079	0.067	0.058	0.051	0.046
Simulated Nos	0.301	0.176	0.126	0.097	0.078	0.067	0.058	0.051	0.046

The following section may be skipped on first reading but for the more curious reader, it gives details of the standard inverse transform method for generating a random variate for the case of a logarithmic distribution.

Generating Benford numbers

In equation (1), a working definition of Benford’s law, it is implicit that the proportions may be expressed as probabilities. Thus, the equation may be recast as:

$$P(D=d)=\log_{10}\left(1+\frac{1}{1+d}\right) \qquad (6)$$

where D is the probability that any digit will be d, d E(1,9). Actually, it is easy to show that equation (6) above holds for any base, not just 10, but in the interests of simplicity, base 10 will continue to be used here.

Equation (6) sums to 1 as a true probability distribution must do: the demonstration of this proceeds as follows:

The probability that the first digit D = d is given by

$$P(D=d)=\sum_{d=1}^{10-1}\log_{10}(\frac{d+1}{d})=\log_{10}\prod_{d=1}^{9}\frac{d+1}{d} \qquad (7)$$

$$\log_{10}\prod_{d=1}^{9}\frac{d+1}{d}=\log_{10}\frac{(9+1)!}{(10-1)!}=1 \qquad (8)$$

which confirms equation (6) as a probability density function. Knowing the probability density function (pdf) makes it possible to get the cumulative distribution function (cdf), which is obtained as follows:

$$P(D\leq d)=\sum_{1\leq d'\leq d}P(D=d')=\sum_{1\leq d'\leq d}\log_{10}\frac{d'+1}{d'}=\log_{10}\left(\prod_{1\leq d'\leq d}\left(\frac{d'+1}{d'}\right)\right) \qquad (9)$$

and then

$$\log_{10}\left(\prod_{1\leq d'\leq d}\left(\frac{d'+1}{d'}\right)\right) = \log_{10}\left(\frac{2}{1}\cdot\frac{3}{2}\cdots\frac{d+1}{d}\right) = \log_{10}\left(\frac{(d+1)!}{d!}\right) = \log_{10}(d+1) \tag{10}$$

so that the distribution function is

$$F(x) = P(X \leq x) = \log_{10}(d+1) \tag{11}$$

for $x = 1, 2, \ldots, 9$

Then the usual inverse transform method given for a Benford variate X is

$$X = \lceil 10^{U} - 1 \rceil \tag{12}$$

whence X ← $\lfloor 10^{U} \rfloor$; this is the procedure implemented in Benford.R from Appendix (iii).

Scaling

A special feature of data sets that obey Benford's law is the following: multiplying a set of numbers that obey Benford's law by some constant number will produce another set of numbers that also obeys Benford's Law. Whatever the constant, the new, different, numbers will also obey the law.

Consider the following small example. Suppose a set of numbers is (1.3, 4.5, 6.2, 2.4, 8.0, 3.0) with leading digits in bold type. Multiplying the set by, say, 1.5, gives a new set. Using a calculator will show that the new leading digits are (1, 6, 9, 3, 1, 4). The numbers have changed, some leading digits disappear, others reappear; so in a large set, it is plausible that the proportions of leading digits might remain the same.

For a demonstration, run the program Fib.R, choosing again to calculate 1000 numbers. This time a small positive constant (= 1) is to be entered; it is to multiply (scale) the Fibonacci numbers. So, for example, the input might be Fib(1000, 1.5). The output, after making allowance for round off error, is almost exactly the same as in the Table 3 Benford values.

This unchangeable feature of Benford's Law is described as scalability or scale invariance. While the above program is only a demonstration, Berger & Hill (2011) gave a formal proof of this characteristic based on manipulations in a σ algebra (but that is beyond the scope of this primer). Further of interest is this fact: Pinkham (1961) had already shown that there can be no other scale invariant distribution of first digits: that is, Benford's is the only one.

A consequence of the scaling property is that physical measurements of natural phenomena in any set of units will obey Benford's Law. So, for the previously mentioned data set of world river lengths, it doesn't matter whether the units are miles or kilometres: nature doesn't discriminate - scaling operates.

Number invention and tampering

One of the first in the field of applications to finance was Nigrini (1999), who, in 1993, was a young accountant. He gives details of an Arizona USA fraud trial in which he was involved. A company employee, in the course of his work, had paid random amounts totalling about $2,000,000 into his own bank accounts. Random was the problem - for the employee, of course; but not for the Court, which compared actual digit occurrences with Benford's Law, and brought down a finding of guilty of fraud. Nigrini's paper is aptly titled 'I've Got Your Number'!

How forensic finance investigation operates will now be illustrated by a basic level example. Table 6 below exhibits first digit data from a file of 2525 entries, containing falsified entries, as quoted in Nigrini (2008), and displayed in Figure 3, where there is an obvious divergence from the Benford values.

Table 6: False Data

Digit	1	2	3	4	5	6	7	8	9
Number	668	381	256	244	235	212	208	179	142

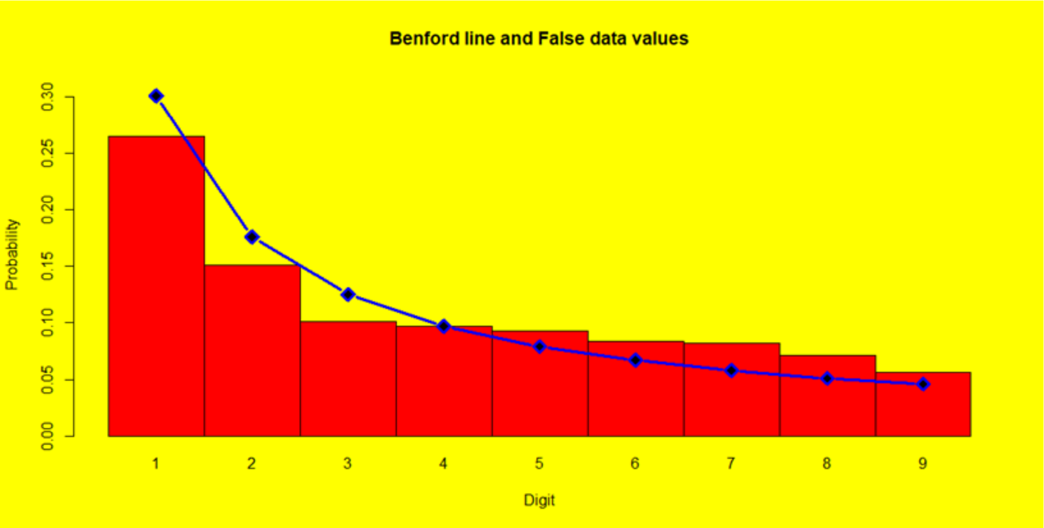


Figure 3: The Benford line and False data values.

The data in Figure 3 is an example only. For real-life data, ascertaining Benford compliance often depends on subsequent Z-tests and ChiSquare tests.

Further digits

Benford probabilities apply not only to the first digits of numbers, but also to second, third, and fourth digits, with a probability law which is more complicated than the one used so far; but a log10 distribution still applies; additionally, there is a term of the $(1 + \frac{1}{1+d})$ type.

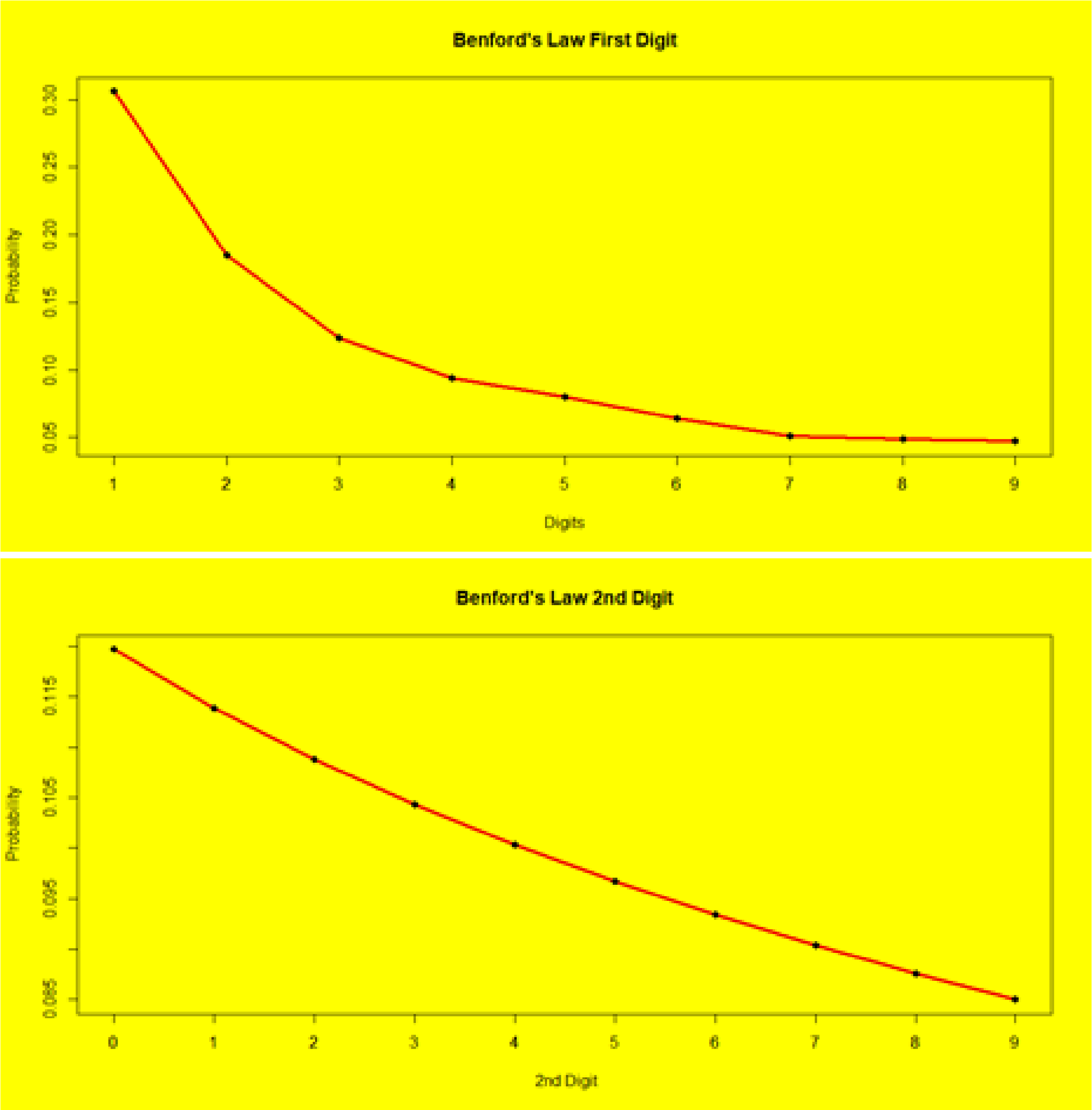
A little preliminary explanation may be useful. Suppose 4 to be a second digit: it can occur as 14, 24, 34, 44,…, 94 so the probability of a 4 occurring must involve a sum of probabilities;

and likewise for the other digits out of (1, . . . ,9). Also, there must be a 10 multiplying any second digit, to ensure that it is fixed in the second decimal place. Finally, 0, which was not admissible as a first digit, can occur as a second digit. Putting all this together results in the probabilities of occurrence of digits first and higher:

$$P(d_1, d_2, ...dm) = \log_{10}\left(1+\frac{1}{\sum_{j=1}^{m}\frac{1}{10^{m-j}dj}}\right) \qquad (13)$$

where P is probability; d_1, d_2, ...,d_m, are digits with $d_j \geq 2$, and m is an integer 0 → 9.

The first digit place is where the distribution of Benford's Law differs the most from the uniform random distribution; subsequent digit probabilities tend more and more towards the uniform. The start of the 'becoming more uniform trend is demonstrated with the probabilities of the first and second digit in Figure 4(a), 4(b) below:



(a) First Digit (b) Second Digit

Figure 4: First and Second Digit

Amongst the many specialized applications of multiple digit analysis, two are mentioned briefly below.

(i) Finance

Among the different general levels of digit analysis, the test for the first digit is the most effective in pointing to suspicious data. It also plays a role in deciding on the size of a sample considered for further investigation, an important factor for auditing cost. Different tests have different functions: one set of five tests, following ACFE (2020), is as listed below: (a) The first digit test (b) The second digit test (c) The first and second digit test (d) The first three digits test (e) The last two digits test. It is impossible to mention here the huge spread of applications - basically any transaction, government or private where money may change hands.

(ii) Election fraud

The second digit test - 2BL for short - has played a role in detecting figures that may suggest election fraud. While an enormous amount has been written on this topic, involving specific election results from particular countries (see Mebane, Walter R.Jr. for examples), academic debate on the application of 2BL to elections is ongoing. The existence of uncertainty at that level of discussion is a clear signal for the conclusion of a primer level understanding of Benford.

Conclusion

The understandings gained by working through this primer should be adequate to progress on to one of the specialized areas of application of Benford’s law. These are far too numerous to mention, but a few are physics, computer science, scientific data quality control, digital imaging forensics, and biology. Their number is constantly expanding.

Appendices

(i) Fib.R

Fibonacci nos. & Binet’s Approximation. Fib(n, k = 1) generates the first n Fibonacci numbers and multiplies them by constant k (defaultvalue = 1)

```
Fib <- function(n, k = 1)
{ #------ begin function
  # fd extracts first digits
  fd <- function(x) {
    a = log10(x) %% 1; floor(10^a) }
  # nos holds generated numbers
  nos <- c() ; s = sqrt(5)
  for(i in 1:n){
   nos[i] =1/s * ((1 + s)/2)^i}
  ben = fd(k * nos) ;
  round(table(ben)/n, 3)
} #------- end function-
```

Examples

benf = Fib(1000) generates the first 1000 Fib.nos
benf = Fib(1000, 1.5) generates the first 1000 Fib.nos multiplied by 1.5

(ii) My Deposit.R

My Deposit.R takes the arguments amount and interest rate. It outputs times and fractional times for digit d to become d + 1

```
My_Deposit <- function(amount, rate){ # begin function
dep = c(rep(0,9)); R =log10(1 + rate/100)
for(d in 1:9){dep[d]= log10((d + 1)/d) *1/R}
dep = round(dep,3)
pc.time = round(dep/sum(dep),3)
out = list("time_to_next_digit_up" = dep,
"time_as_fraction _of_total_time: 1 -> 9" = pc.time)
return(out)
} # end function
```

Example: My_Deposit(10,5)

(iii) Benford.R

Benford.R generates numbers obeying Benford’s Law

```
# N.B. increments are drawn from R’s Uniform Distribution
# Benford.R
# Final version of Benford numbers
N = 500000; # number of replications
x = floor(10^runif(N)); # digits
benf = table(x)/N
round(benf,3)
```

Output: 500000 first digits sorted into bins (1,. . . , 9)

References

Berger A.& Hill T.P. (2011). A basic theory of Benford's Law. *Probability Surveys, 8,* 1-126.

Benford, F. (1938). The law of anomalous numbers. *Proceedings of the American Philosophical Society 78*, 551-572.

Cinelli, C. (2018). Benford Analysis.R, Retrieved 1 November 2021, from http://github.com/carloscinelli/benford.analysis.

Lanham S. W. (2019). Analyzing big data with Benford's Law. *American Journal of Business Education, 12*(2), 33-42.

Leemis L. (2018). *Probability,* Lightning Source: USA.

Miller S.J. (2015). *A quick introduction to Benford's Law*, Princeton University Press: USA.

Newcomb, R. (1881). Note on the frequency of use of the different digits in natural numbers *American Journal of Mathematics, 4,* 39–40.

Nigrini, M. (2008). The problem of false negative results in the use of digit analysis *Journal of Applied Business Research, 24*(1), 17-26.

Nigrini, M. (1999). I've got your number *Journal of Accountancy, 187*(5), 79-83.

Pinkham, R. S. (1961). Ann. Math. Stats. 32: 1223-1230 R, https://www.r-project.org

Stoessiger, R. (2013). Benford's Law and why the integers are not what we think they are: A critical numeracy of Benford's Law *Australian Senior Mathematics Journal, 27*(1), 29-46.

US Bureau of Labor Statistics (2021). *Occupational employment and wage statistics* Retrieved 1 November 2021, from https:/www.bls.gov/oes/current/oessrest.htm.