Feb 21, 2006

Catching cheaters with Benford's Law

Catching cheaters with Benford's Law

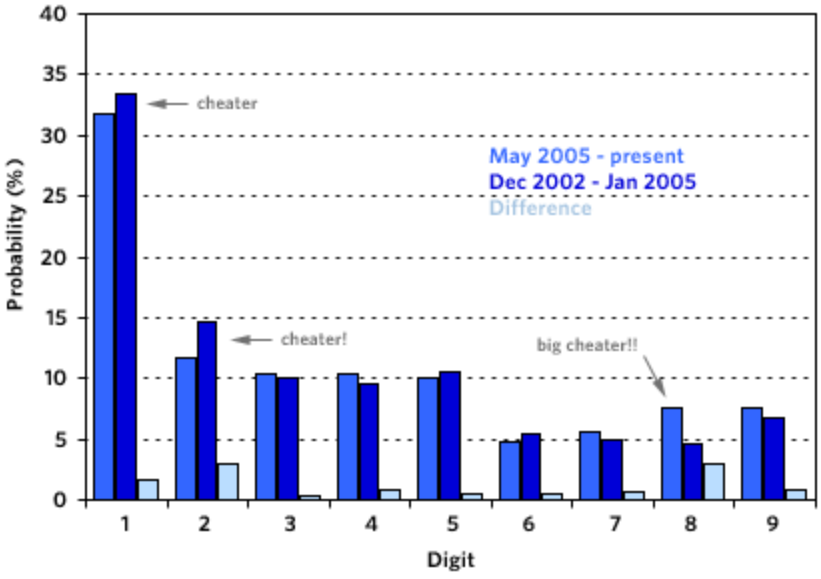
Benford's Law describes a curious phenomenon about the counterintuitive distribution of numbers in sets of non-random data:

> A phenomenological law also called the first digit law, first digit phenomenon, or leading digit phenomenon. Benford's law states that in listings, tables of statistics, etc., the digit 1 tends to occur with probability ~30%, much greater than the expected 11.1% (i.e., one digit out of 9). Benford's law can be observed, for instance, by examining tables of logarithms and noting that the first pages are much more worn and smudged than later pages (Newcomb 1881). While Benford's law unquestionably applies to many situations in the real world, a satisfactory explanation has been given only recently through the work of Hill (1996).

I first heard of Benford's Law in connection with the IRS using it to detect tax fraud. If you're cheating on your taxes, you might fill in amounts of money somewhat at random, the distribution of which would not match that of actual financial data. So if the digit "1" shows up on Al Capone's tax return about 15% of the time (as opposed to the expected 30%), the IRS can reasonably assume they should take a closer look at Mr. Capone's return.

Since I installed Movable Type 3.15 back in March 2005, I have been using its "post to the future" option pretty regularly to post my remaindered links...and have been using it almost exclusively for the last few months[1]. That means I'm saving the entries in draft, manually changing the dates and times, and then setting the entries to post at some point in the future. For example, an entry with a timestamp like "2006-02-20 22:19:09" when I wrote the draft might get changed to something like "2006-02-21 08:41:09" for future posting at around 8:41 am the next morning. The point is, I'm choosing basically random numbers for the timestamps of my remaindered links, particularly for the hours and minutes digits. I'm "cheating"... committing post timestamp fraud.

That got me thinking...can I use the distribution of numbers in these post timestamps to detect my cheating? Hoping that I could (or this would be a lot of work wasted), I whipped up a MT template that produced two long strings of numbers: 1) one of all the hours and minutes digits from the post timestamps from May 2005 to the present (i.e. the cheating period), 2) and one of all the hours and minutes digits from Dec 2002 - Jan 2005 (i.e. the control group). Then I used a PHP script to count the numbers in each string, dumped the results into Excel, and graphed the two distributions together. And here's what they look like, followed by a table of the values used to produce the chart:



Digit	5/05-now	12/02-1/05	Difference
1	31.76%	33.46%	1.70%
2	11.76%	14.65%	2.89%
3	10.30%	9.96%	0.34%
4	10.44%	9.58%	0.86%
5	10.02%	10.52%	0.51%
6	4.83%	5.40%	0.57%
7	5.66%	4.96%	0.70%
8	7.62%	4.65%	2.97%
9	7.60%	6.81%	0.79%

As expected, 1 & 2 show up less than they should during the cheating period, but not overly so[2]. The real fingerprint of the crime lies with the 8s. The number 8 shows up during the cheating period ~64% more than expected. After thinking about it for awhile, I came up with an explanation for the abundance of 8s. I often schedule posts between 8am-9am so that there's stuff on the site for the early-morning browse and I usually finish off the day with something between 6pm-7pm (18:00 - 19:00). Not exactly the glaring evidence I was expecting, but you can still tell.

The obvious next question is, can this technqiue be utilized for anything useful? How about detecting comment, trackback. or ping spam? I imagine IPs and timestamps from these types of spam are forged to at least some extent. The difficulties are getting enough data to be statistically significant (one forged timestamp isn't enough to tell anything) and having "clean" data to compare it against. In my case, I knew when and where to look for the cheating...it's unclear if someone who didn't know about the timestamp tampering would have been able to detect it. I bet companies with services that deal with huge amounts of spam (Gmail, Yahoo Mail, Hotmail, TypePad, Technorati) could use this technique to filter out the unwanted emails, comments, trackbacks, or pings...although there's probably better methods for doing so.

[1] I've been doing this to achieve a more regular publishing schedule for kottke.org. I typically do a lot of work in the evening and at night and instead of posting all the links in a bunch from 10pm to 1am, I space them out over the course of the next day. Not a big deal because increasing few of the links I feature are time-sensitive and it's better for readers who check back several times a day for updates...they've always got a little something new to read.

[2] You'll also notice that the distributions don't quite follow Benford's Law either. Because of the constraints on which digits can appear in timestamps (e.g. you can never have a timestamp of 71:95), some digits appear proportionally more or less than they would in statistical data. Here's the distribution of digits of every possible time from 00:00 to 23:59:

1 - 25.33
2 - 17.49
3 - 12.27
4 - 10.97
5 - 10.97
6 - 5.74
7 - 5.74
8 - 5.74
9 - 5.74