by Jeffreys Copeland and Haemer

SCOTT ROBERTS

Animal crackers in my soup,
Monkeys and rabbits loop
the loop, Gosh, oh, gee, but
I have fun!
– Shirley Temple ("Curley-top," Ted Koehler, Edward Heyman and Irving Caesar)

I have no faith in anything short of actual measurement and the rule of three.
– Charles Darwin

Real Cookies

In the last two issues, we looked at eXtreme Testing: the practice of deciding what results your code should produce before you start coding. This month, we'll do precisely the opposite. We'll use the UNIX command line as a tool for rapid Exploratory Data Analysis, which means, approximately, "looking at data without much notion of what we should find to see if we can discover something interesting."

Cookies

We have, for many years, kept huge supplies of cookies in our office that empty and refill at least weekly. We graze constantly. We have no idea why we're not as big as houses; we tentatively conclude that, for us, writing software is a high-calorie activity. Drinking three pots of coffee a day probably keeps our metabolic rate up, too.

About six months ago, we decided that the crowd of visitors in our office was keeping us from getting work done, so we put all our cookies in a colorful cookie tin and gave it to Paul, in the office next door. Paul caught on more quickly than we did and moved the tin into the lab after only a few weeks.

This hasn't, however, left our office cookie-free. On a low shelf of one bookcase, we keep a supply of animal crackers. We put them on a low shelf for the same reason grocery stores do: they're not for adults.

In case you haven't noticed, every store stocks its animal crackers on low shelves, at child-height. That appears to be Nabisco's entire, and wildly successful, marketing strategy; we doubt that there is any man, woman, or child in America who hasn't eaten them and doesn't recognize the packages instantly, but we've never, ever, seen an ad for animal crackers.

(And yes, we're talking specifically about Nabisco's product, Barnum's Animals Crackers. You'll occasionally see other, animal-shaped cookies marketed, but never for long.)

So, since you've eaten them (unless you're reading this in a country that doesn't have animal crackers), can you list the animals? Try doing it without looking. (If you're not up to that, try naming the seven dwarfs.)

There are 18: bear (sitting), bear (walking), bison, camel, cougar, elephant, giraffe, gorilla, hippopotamus, kangaroo, lion, monkey, rhinoceros, seal, sheep, tiger, hyena, and zebra.

They don't correspond to the animals pictured on the box, so if you tried to cheat by using those, you're out of luck. We eventually got the list above from Nabisco's Web site, but not until long after we'd investigated the problem ourselves.

To start, we just opened a box and looked to see what was there. Here's the first box we tried:

1 bear, 1 bison, 2 gorillas, 1 hippo,
3 hyenas, 1 kangaroo, 1 lion,
1 monkey, 2 seals, 2 tigers, 2 zebras.

(We'll use "bear" instead of "bear,

standing," and "polar bear" instead of "bear, walking," because that's what they look like to us. Besides, they're easier to say and type. Oh, and in our opinion, the "hyena" looks more like a wolf.)

Did Nabisco make three times as many hyenas as hippos, or was this just sampling variation? And if it was sampling variation, were there animals we weren't seeing? More investigation was required. We ate the first sample and opened another box:

1 bear, 3 camels, 1 cougar, 1 gorilla, 2 hippos, 3 lions,
1 polar bear, 2 rhinos, 2 seals, 1 sheep, 1 tiger, 1 zebra.

It looks like sampling variation, since the second box had no hyenas, but did have several new animals–camels, a cougar, a polar bear, rhinos, and a sheep.

Ah, a statistics problem. Two samples is too small to conclude anything about the distribution, so we ate the second box and tried a third:

2 bears, 2 bisons, 3 camels, 1 elephant, 1 gorilla, 2 hippos,
3 hyenas, 4 monkeys, 1 tiger, 1 zebra.

Only one new animal this time–the elephant–but who knows what that means without a lot of math?

We decided this was early enough in the investigative phase to make more data collection imperative. To prevent accidental contamination of the next set, we ate the third data set and opened a fourth box.

1 bear, 2 bison, 1 elephant, 2 giraffes, 1 hippo, 2 hyenas
1 kangaroo, 2 lions, 2 monkeys, 2 polar bears, 1 rhino,
2 sheep, 1 tiger.

No new animals; at this point, we probably had a complete list.

Of course, that's only if all the animals are equally frequent. If some are far more numerous than others, we could still be missing rare ones. We had to look at the numbers.

Ignoring a passing co-worker, who ate the tiger and one of the sheep while remarking that anyone who sorts and counts his animal crackers is far too compulsive, we made a pot of coffee, rolled up our sleeves, and began to write code. We also ate the rest of the fourth sample, to clear the way for more test data.

Counting

After putting our data into files, in the simple, two-column format, `animal number`, we began running experiments.

• How many animals are in each box? We love `awk` for simple arithmetic on columnar data.

```
$ awk '{n += $2}; END {print n}' box_*
```

Our four boxes had 17, 18, 20, and 20 cookies. Clearly, there isn't a machine counting how many cookies each box gets; perhaps boxes are filled by weight and some animals are heavier than others.

Just as clearly we needed more test data. After another pot of coffee, we settled for 16 data sets, both because we're attracted to numbers like 2^{22} and we didn't think we could eat 65,536 ($2^{2^{22}}$) boxes of animal crackers before our column deadline.

This quickie:

```
$ awk '{n += $2} END {print n}' box_* |
> sort -n | uniq -c |
> awk '{print $2, $1}'
```

gives us the distribution of cookies-per-box:

```
17 1
18 3
19 5
20 5
21 2
```

So, the median number of cookies-per-box is 19.5, and the distribution looks quite symmetrical.

• How many kinds of animals are in each box? Here again, `awk` makes our life easy. `wc -l` gives us, in two columns, the number of species per box, plus a total.

```
$ wc -l box*
      10 box_1
      11 box_10
      12 box_11
      11 box_12
      11 box_13
      12 box_14
      10 box_15
      13 box_16
      12 box_2
      13 box_3
      12 box_4
      14 box_5
      12 box_6
      13 box_7
      13 box_8
      12 box_9
      191 total
```

Using `awk`'s ability to filter lines through regular expressions gives us a quick answer from the command line.

```
$ wc -l box_* | awk '/total/ {print $1/(NR-1)}'
11.9375
```

• Is Nabisco species-ist? Do all animals occur with equal frequency, or are some more heavily represented than others (say, because they're lighter)?

We'll just sum the number of each type:

```
$ awk '{n[$1] += $2};
> END {for (i in n) print i, n[i]}' box_* |
```

```
> sort -n +1
kangaroo 9
panther 12
bear 13
giraffe 15
hippo 15
lion 15
bison 16
polar bear 16
tiger 16
zebra 16
elephant 19
gorilla 20
hyena 20
rhino 20
seal 21
sheep 21
camel 22
monkey 22
```

Looks like there may really be more monkeys and camels than kangaroos.

(The O'Reilly "Camel" book is "Programming Perl," ISBN 0-596-00027-8. The O'Reilly "Monkey" books are "Essential Windows NT System Administration," ISBN 1-56592-274-3, and "Win32 API Programming with Visual Basic," ISBN 1-56592-631-5. The O'Reilly "Kangaroo" book is "Power Programming with RPC," ISBN 0-937175-77-3. Coincidence? Sure, that's one theory.)

Very little work rearranges this into a distribution.

```
$ awk '{n[$1] += $2};
>    END {for (i in n) print n[i]}' box_* |
> sort | uniq -c |
> awk '{print $2, $1}' | sort -n
9 1
12 1
13 1
15 3
16 4
19 1
20 3
21 2
22 2
```

Translation: one animal (the kangaroo) appeared nine times, three, 15 times, two, 22 times, etc. Besides the low frequency of kangaroos, there also may be a distinction between the rarer animals, like zebras, and the commoner ones, like gorillas.

This result has been observed, in a different context, by another author: "All animals are equal, but some animals are more equal than others." – "Animal Farm," George Orwell

Hypothesis Testing

Ah, but are the animals really present in different frequencies? If we flipped a coin 100 times and got all heads, we'd look to see if we had a two-headed coin. But 53 heads and 47 tails wouldn't even make us suspect a bias.

What's more, in a perfect world, if all the animals were created equal–or at least equally–we'd expect 308/18 = 17.11 camels and 17.11 kangaroos. We know that wouldn't happen–animals don't come in fractions.

(Yes, we got broken cookies. For our counts, we pieced them back together. We were pleased, if a little surprised, to discover that this usually works. Nabisco doesn't fill boxes with random pieces of broken cookies.)

So, is Nabisco really making the same number of each animal? Did we just get these inequalities by the luck of the draw? To decide, we needed some measure of how different our observations are from the ideal–equal frequencies of all animals. We also needed a theoretical probability distribution of that measurement.

If you've taken too many statistics courses or science labs, you'll immediately suspect that the right tool for the job is the χ^2 (chi-squared) test. We have, and we did.

In a χ^2 test, the measure for overall difference is

$$\chi^2 = \Sigma \frac{(observed\text{-}expected)^2}{expected}$$

Notice that this makes some sense. Squaring each difference makes five too few panthers (-5) count the same as five

too many monkeys (+5). Dividing by an expected number "normalizes" things. Since we only expected 17 kangaroos, eight too few is more impressive than if we'd expected 600.

Here's a quick program to calculate the statistic:

```
#!/usr/bin/perl -w
# $Id: chi2,v 1.2 2000/08/04 19:55:50 jsh Exp $

use strict;
my ($sum, $ssd, %n);

while (<>) {
 my ($key, $val) = split;
 $n{$key} = $val;
 $sum += $val;
}
my $exp = $sum/(keys %n);

foreach (keys %n) {
 $ssd += ($n{$_} - $exp)**2
}

print $ssd/$exp, "\n";
```

For our data, we got $\chi^2 = 13.7$. Okay … is this number too big, too small, or, Goldilocks-like, just right?

Like *sin()* and *cos()*, χ^2 usually sends us in search of CRC handbooks. In this case, though, a back-of-the-envelope calculation will do.

Anyone who's spent enough time drinking beer and eating animal crackers with statisticians knows that a χ^2 distribution with N degrees of freedom has $\mu = N$ and $\sigma^2 = 2N$ (where, of course, μ is the mean, and σ the standard deviation). A χ^2 statistic with 18 categories that sum to a fixed total has 17 degrees of freedom. We expect a χ^2 of about 17, and would not even blink hard until we get χ^2 values below about five or above about 23 ($\mu \pm 2\sigma$).

What we see is within a standard deviation of the mean; we have no particular reason to believe that Nabisco makes more of any one animal than of any other.

Manufacturer's Claims

Okay, now that we know what the boxes actually hold, let's go see what Nabisco says, at `http://www.nabiscoworld.com/Barnums/ba_info.htm`:

"There are 17 different animals in 18 different forms with the bear having two shapes, sitting and walking. There is an average of 15 different animals in each box with 2 or 3 of the same species to total 22 or 23 animals in a box."

Well, no. Boxes typically total 19 or 20 cookies, in an average of just under a dozen types.

Nabisco's Web site claims their boxes are about 15% fuller than they are, and that they average fully 25% more types-per-box than they actually do.

Oops.

On the other hand, if they were that much fuller, we would be, too.

We offer free beer and animal crackers to readers willing to come consume them with us while working out the statistical significance of the deviation between Nabisco's claims and our observations.

Until then, or next time, we wish you happy tails, er, trails.

Loose Ends

A few interesting bits of data have crossed our paths since our June column on Penrose tilings, in which we tried to figure out how to pattern the floor in the foyer of Copeland's new house.

First, our summer beach reading was Neal Stephenson's excellent "Cryptonomicon." Imagine our surprise when one of Stephenson's heroes visits a well-off friend in Seattle and discovers, "Even the pavement under his feet [was] some kind of custom-made mosaic of Penrose tiles." This just goes to prove that great minds think alike. Or perhaps that we sometimes, accidentally, think like great minds.

We didn't think we could eat 65,536 boxes of animal crackers before our column deadline.

Second, and more important, we had a long note from reader Jim Homan in Ft. Collins, CO, pointing out our mathematical errors. It seems that our original understanding was insufficient, and we used a few patterns that aren't actually valid Penrose tilings. For example, Penrose tiles aren't allowed to meet "nose to nose," nor are different types of tiles allowed to meet "shoulder to shoulder." This means that the composite motion in our `{co addtop} PostScript macro` always makes invalid tilings, and some of the other macros can be misused to do so. Jim pointed us to a 1977 article by Martin Gardner at `http://scientium.com/drmatrix/progchal.htm`, which is a much better treatment than the one we provided. We've corrected the software on our Web site.

Jeffrey Copeland *(*`copeland@alumni.caltech.edu`*) is currently living in the Pacific Northwest, where he spends his time writing UNIX software in a large development organization and fighting damp rot.*

Jeffrey S. Haemer *(*`jsh@usenix.org`*) works at Minolta-QMS Inc. in Boulder, CO, building laser printer firmware. Before he worked for QMS, he operated his own consulting firm and did a lot of other things, like everyone else in the software industry.*

Note: The software from this and past Work columns is available at `http://alumni.caltech.edu/~copeland/work` *or alternately at* `ftp://ftp.cpg.com/pub/Work`.