

Re: Probability Problem

Source: <http://coding.derkeiler.com/Archive/Python/comp.lang.python/2006-04/msg03787.html>

- *From:* Elliot Temple <curi@xxxxxxx>
 - *Date:* Tue, 25 Apr 2006 00:09:08 -0700
-

I think I got it. I noticed my code is essentially the same as Tim Peter's (plus the part of the problem he skipped). I read his code 20 minutes before recreating mine from Alex's hints. Thanks!

```
def main():
    ways = ways_to_roll()
    total_ways = float(101**10)
    running_total = 0
    for i in range(1000-390+1):
        j = i + 390
        running_total += ways[i] * ways[j]
    print running_total / total_ways**2
    print ways[:10]

def ways_to_roll():
    result = [1]
    for i in xrange(10):
        result = combine([1] * 101, result)
    return result

def combine(a, b):
    results = [0] * (len(a) + len(b) - 1)
    for i, ele in enumerate(a):
        for j, ele2 in enumerate(b):
            results[i+j] += ele * ele2
    return results

main()
# output: 3.21962542309e-05 and
# [1, 10, 55, 220, 715, 2002, 5005, 11440, 24310, 48620]
# 3.21962542309e-05 is 32 out of a million
```

On Apr 24, 2006, at 9:14 PM, Alex Martelli wrote:

Elliot Temple <curi@xxxxxxx> wrote:

Re: Probability Problem

On Apr 24, 2006, at 8:24 PM, Alex Martelli wrote:

Lawrence D'Oliveiro
<ldo@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx> wrote:

In article
<mailman.4949.1145931967.27775.python-list@xxxxxxxxxx>,
Elliot Temple <curi@xxxxxxx> wrote:

Problem: Randomly
generate 10 integers from
0–100 inclusive, and sum
them. Do that twice. What is
the probability the two sums
are 390
apart?

I think the sum would come close to a
normal distribution.

Yes, very close indeed, by the law of large numbers.

However, very close (in a math course at least) doesn't get
the cigar.

You can compute the requested answer exactly with no
random number
generation whatsoever: compute the probability of each
result from
0 to
1000, then sum the probabilities of entries that are exactly
390
apart.

That was the plan, but how do I get the probability of any given
result? (in a reasonable amount of time)

BTW I'm not in a math course, just curious.

OK, I'll trust that last assertion (sorry for the hesitation, but it's
all too easy to ``help" somebody with a homework assignment and
actually end up damaging them by doing it FOR them!–).

Re: Probability Problem

I'm still going to present this in a way that stimulates thought, rather than a solved problem — humor me...!-)

You're generating a uniformly distributed random number in 0..100 (101 possibilities), 10 times, and summing the 10 results.

How do you get a result of 0? Only 1 way: 0 at each attempt — probability 1 (out of 1010 possibilities).

How do you get a result of 1? 10 ways: 1 at one attempt and 0 at each of the others — probability 10 (again in 1010's;-).

How do you get a result of 2? 10 ways for '2 at one attempt and 0 at each of the others', plus, $10 \cdot 9/2$ ways for '1 at two attempts and 0 at each of the others' — probability 55 (ditto).

...and so forth, but you'd rather not work it out...

So, suppose you start with a matrix of 101 x 10 entries, each of value 1 since all results are equiprobable (or, 1/1010.0 if you prefer;-).

You want to compute the number in which you can combine two rows. How could you combine the first two rows (each of 101 1's) to make a row of 201 numbers corresponding to the probabilities of the sum of two throws?

Suppose you combine the first entry of the first row with each entry of the second, then the second entry of the first row with each entry of the second, etc; each time, you get a sum (of two rolls) which gives you an index of a entry (in an accumulator row starting at all zeros) to increment by the product of the entries you're considering...

Can you generalize that? Or, do you need more hints? Just ask!

Alex

--

<http://mail.python.org/mailman/listinfo/python-list>

-- Elliot Temple

<http://www.curi.us/blog/>

.