Genome Sciences 373
Genome Informatics

Quiz Section 9
May 26 2015
Random numbers: so intuitive but yet...

DILBERT

TOUR OF ACCOUNTING

OVER HERE WE HAVE OUR RANDOM NUMBER GENERATOR.

NINE NINE NINE NINE NINE

ARE YOU SURE THAT’S RANDOM?

THAT’S THE PROBLEM WITH RANDOMNESS: YOU CAN NEVER BE SURE.
Random numbers: so intuitive but yet…

```
int getRandomNumber()
{
    return 4;  // chosen by fair dice roll.
    // guaranteed to be random.
}
```

People don’t necessarily “get” randomness very well

- “Why does my iTunes shuffle play the same band twice in a row?”
- “Why did Solitaire put all of the aces at the bottom? That’s not very random.”

xkcd.com/221/
Random numbers: so intuitive but yet...

There is no “test” to determine if a number is really random

Most computer-generated random numbers are actually *pseudo-random*

But, most pseudo-random numbers are “random enough”

In fact, as we’ll see, sometimes we want the predictability
Random numbers: so intuitive but yet...

True random number generators
- Come from physical processes like atmospheric or thermal noise
- Do not repeat periodically
- Not predictable
- Conceptually simple but usually hard to obtain

Pseudo-random number generators
- Generated by algorithms
- Have periods, which may be arbitrarily long
- Can be more or less predictable, depending
- Can be initialized with a particular seed to yield a predictable outcome
Random numbers: so intuitive but yet...

True random number generators

Pseudo-random number generators

source: Bo Allen [boallen.com]
Random numbers: so intuitive but yet...

Defining a “seed”

A seed lets us initialize the random number generator: sort of a “starting point” for the algorithm.

If you know the seed, the sequence of numbers is predictable.

If you don’t know the seed, the sequence is hopefully unpredictable (but still fixed).
Defining a “seed”

A seed lets us initialize the random number generator: sort of a “starting point” for the algorithm.

If you know the seed, the sequence of numbers is predictable.

If you don’t know the seed, the sequence is hopefully unpredictable (but still fixed).
Why would I *not* want predictable numbers?

You’re sending a secret message and you need a code that’s really hard to crack.

You could set the seed to the current time (milliseconds) – hard to guess and maybe “random enough.”
Why would I want predictable numbers?

Let’s say you’re working on a program and you keep hitting a bug that you need to fix.

Let’s say you’re submitting a paper involving simulations, and you want your work to be reproducible.
Random numbers in python

In python, we can import random and then set the seed using `random.seed(my_seed)`.
Random numbers in python

The pseudo-random number generator and its seed applies to all of the functions we’ve looked at:

- random roll of a die
- random float between 0 and 1
- random column from an alignment for bootstrapping
Random numbers in python

We can generate numbers from non-uniform distributions

Normal distribution

```python
>>> random.gauss(0,1)
-0.5909118323344412
>>> random.gauss(0,1)
1.1846605185219836
>>> random.gauss(0,1)
0.9461677129223653
```
Random numbers in python

Other probability distributions (exponential, gamma, etc) have built-in generators in python

What if our distribution of interest doesn’t have a built-in function (like binomial)?
Where do I set the seed if I want to make this reproducable?