Quiz Section Week 8
May 16, 2017

String handling and regular expressions

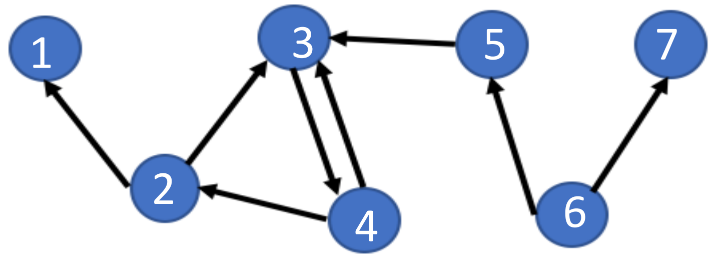
A bit more on generating random numbers

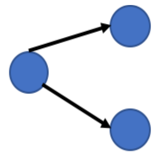
Machine Learning things to think about

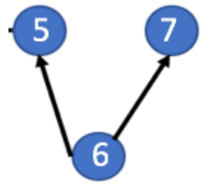
HW 6 problem 2

- Subgraphs vs motifs

Motif = a pattern of connections between nodes
Subgraph = an actual set of nodes and edges







How many subgraphs? Why n-2?

What would the graph have to look like to have that many subgraphs?
Try for a 4 node graph

Homework programming: many of you assumed you know how many lines the file is

```
seq = lines[1]
seq2 = lines[3]
```

- What if you wanted to run your code on a file with 1000 sequences?

```
for line in fin:
        if line[0] == '>':
                headers.append(line.rstrip())
        else:
                seqs.append(line.rstrip())
```

```
count = 0
while count < len(lines):
        headers.append(lines[count])
        seqs.append(lines[count+1])
        count = count + 2
```

More on manipulating strings

- Fasta sequence files are usually formatted as follows:

```
>NC_018723.2 Felis catus isolate Cinnamon breed Abyssinian chromosome A1, Felis_catus_8.0, whole genome shotg
AGAGACTCCAAAATTGGACCCACAAAAGTATGGCCAACTAATCTTTGACAAAGCAGGAAAGA
ATATCCAATGGAAAAAAGACAGTCTCTTTTACAAATGGTGCTGGGAGAACTGGACAGCAACAT
GCAGAAGGTTGAAACTAGACCACTTTCTCACACCATTCACAAAAATAAACTCAAAATGGATAAA
GGACCTGAATGTGAGACAGGAAACCATCAAAACCCTAGAAGAGAAAGCAGGAAAAAAACCTC
TCTGACCTCAGTCGCAGCAATTTCTTACTTGACACATCCCCAAAGGCAAGGGAATTAAAAGCAA
>NC_018724.2 Felis catus isolate Cinnamon breed Abyssinian chromosome A2, Felis_catus_8.0, whole genome shotg
AATGAACTATTGGGACCTCATGAAGATAAAAAAACTTCTGCACAGCAAAGGAAACAATCAACAA
AACTAAAAGGCAACCAACGGAATGGGAAAATACATTTGCAAATGACATATTGGACAAAGGGCTA
GTATCCAAAATCTA
…
>NC_018725.2 Felis catus isolate Cinnamon breed Abyssinian chromosome A3, Felis_catus_8.0, whole genome shotg
…
```

How to read a Fasta file like this one into two lists of strings, one of headers and one of sequences?

Reading in a file with different procedure depending on content of each line

```
fin = open('cat_genome.fasta', 'r')
seqs = []
headers = []
current_seq = ""
for line in fin:
```

Reading in a file with different procedure depending on content of each line

```
fin = open('cat_genome.fasta', 'r')
seqs = []
headers = []
current_seq = ""
for line in fin:
    if line[0] == '>':
        headers.append(line.rstrip())
        seqs.append(current_seq)
        current_seq = ""
    else:
        current_seq = current_seq + line.rstrip()
```

Maybe we want to pull different pieces of information out of the sequence header

```
>NC_018723.2 Felis catus isolate Cinnamon breed Abyssinian chromosome A1, Felis_catus_8.0, whole genor
AGAGACTCCAAAATTGGACCCACAAAAGTATGGCCAACTAATCTTTGACAAAGCAGGAAAGA
ATATCCAATGGAAAAAAGACAGTCTCTTTTACAAATGGTGCTGGGAGAACTGGACAGCAACAT
GCAGAAGGTTGAAACTAGACCACTTTCTCACACCATTCACAAAAATAAACTCAAAATGGATAAA
GGACCTGAATGTGAGACAGGAAACCATCAAAACCCTAGAAGAGAAAGCAGGAAAAAAACCTC
TCTGACCTCAGTCGCAGCAATTTCTTACTTGACACATCCCCAAAGGCAAGGGAATTAAAAGCAA
>NC_018724.2 Felis catus isolate Cinnamon breed Abyssinian chromosome A2, Felis_catus_8.0, whole genor
AATGAACTATTGGGACCTCATGAAGATAAAAAAACTTCTGCACAGCAAAGGAAACAATCAACAA
AACTAAAAGGCAACCAACGGAATGGGAAAATACATTTGCAAATGACATATTGGACAAAGGGCTA
GTATCCAAAATCTA
…
>NC_018725.2 Felis catus isolate Cinnamon breed Abyssinian chromosome A3, Felis_catus_8.0, whole genor
```

- NCBI ID
- Species
- Isolate
- Breed
- Chromosome number

We can use *regular expressions* to analyze patterns in strings

```
import re
line = '>NC_018723.2 Felis catus isolate Cinnamon
breed Abyssinian chromosome A1, Felis_catus_8.0,
whole genome shotgun sequence'


re.match('>NC_', line)


re.findall('chromosome', line)


re.sub('whole genome shotgun', 'WGS', line)
```

But what if we don't always have these characters exactly?

https://docs.python.org/2/library/re.html

Regular expression glossary (incomplete!)

. any character

***** repeated 0 or more times

+ repeated 1 or more times

{n} repeated n times

[A|B] either A or B

[A-Z] any uppercase letter, **[0-9]** any numeric character

^ beginning of line

$ end of line

\ escape (actually search for one of the characters above)

We can use *regular expressions* to analyze patterns in strings

```
import re
line = '>NC_018723.2 Felis catus isolate Cinnamon breed
Abyssinian chromosome A1, Felis_catus_8.0, whole genome
shotgun sequence'


re.match('chromosome.{3}', line)
```

```
'chromosome A1'
```

```
re.sub('[W|w]hole [G|g]enome [S|s]hotgun', 'WGS', line)
```

```
'>NC_018723.2 Felis catus isolate Cinnamon breed Abyssinian chromosome A1, Felis_catus_8.0, WGS'
```

```
re.findall('^>[A-Z]+_[0-9]+\.[0-9]', line)
```

```
'>NC_018723.2'
```

https://docs.python.org/2/library/re.html

We can use *regular expressions* to analyze patterns in strings

```
import re
line = 'ATGGCTATC'
re.match('AT[G|C]', line)
re.findall('AT[G|C]', line)
re.sub('AT[G|C]', 'QQQ', line)
```

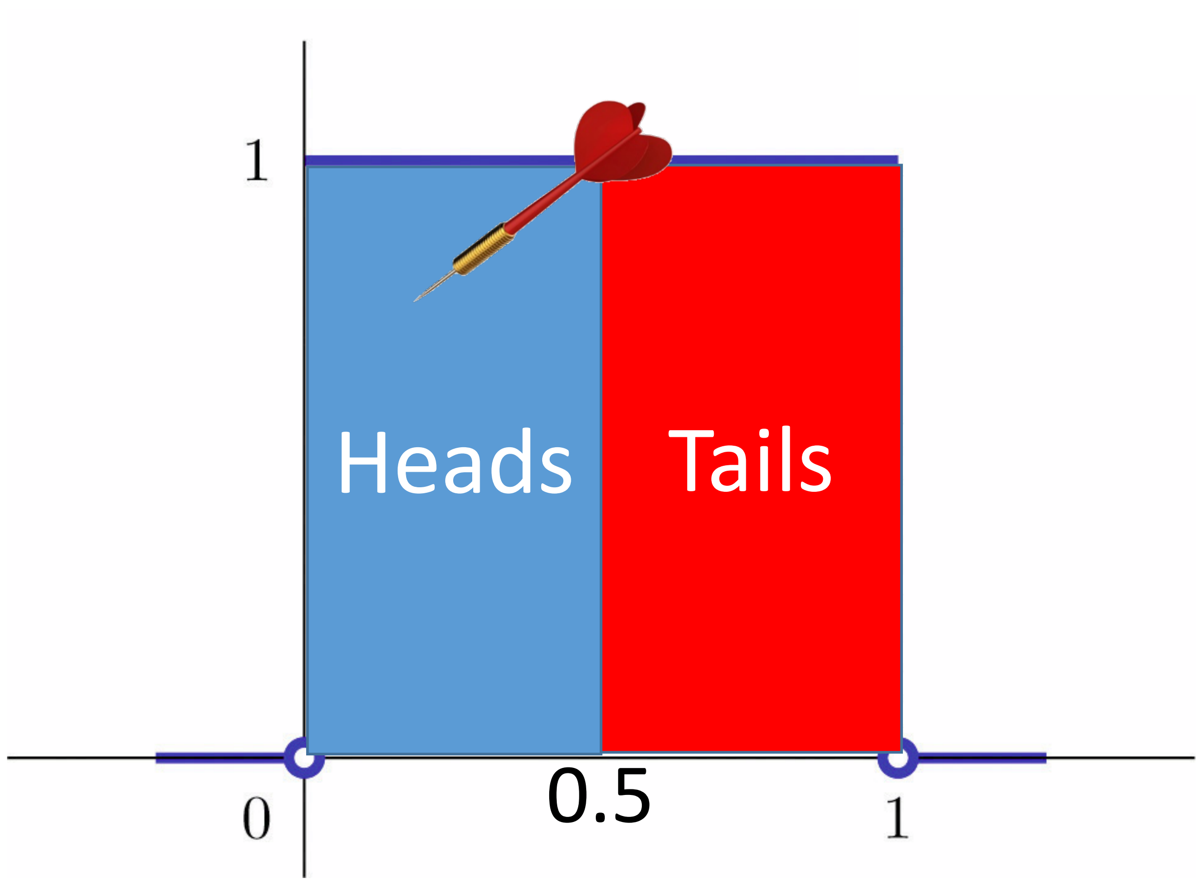
Exercise: use regular expressions to extract the annotation IDs: Felis_catus_8.0

>NC_018723.2 Felis catus isolate Cinnamon breed Abyssinian chromosome A1, Felis_catus_8.0, whole genome shotgun sequence
>NC_018724.2 Felis catus isolate Cinnamon breed Abyssinian chromosome A2, Felis_catus_7.0, whole genome shotgun sequence
>NC_018725.2 Felis silvestrus isolate Cinnamon breed Abyssinian chromosome A3, Felis_silvestris_1.0, whole genome shotgun sequence

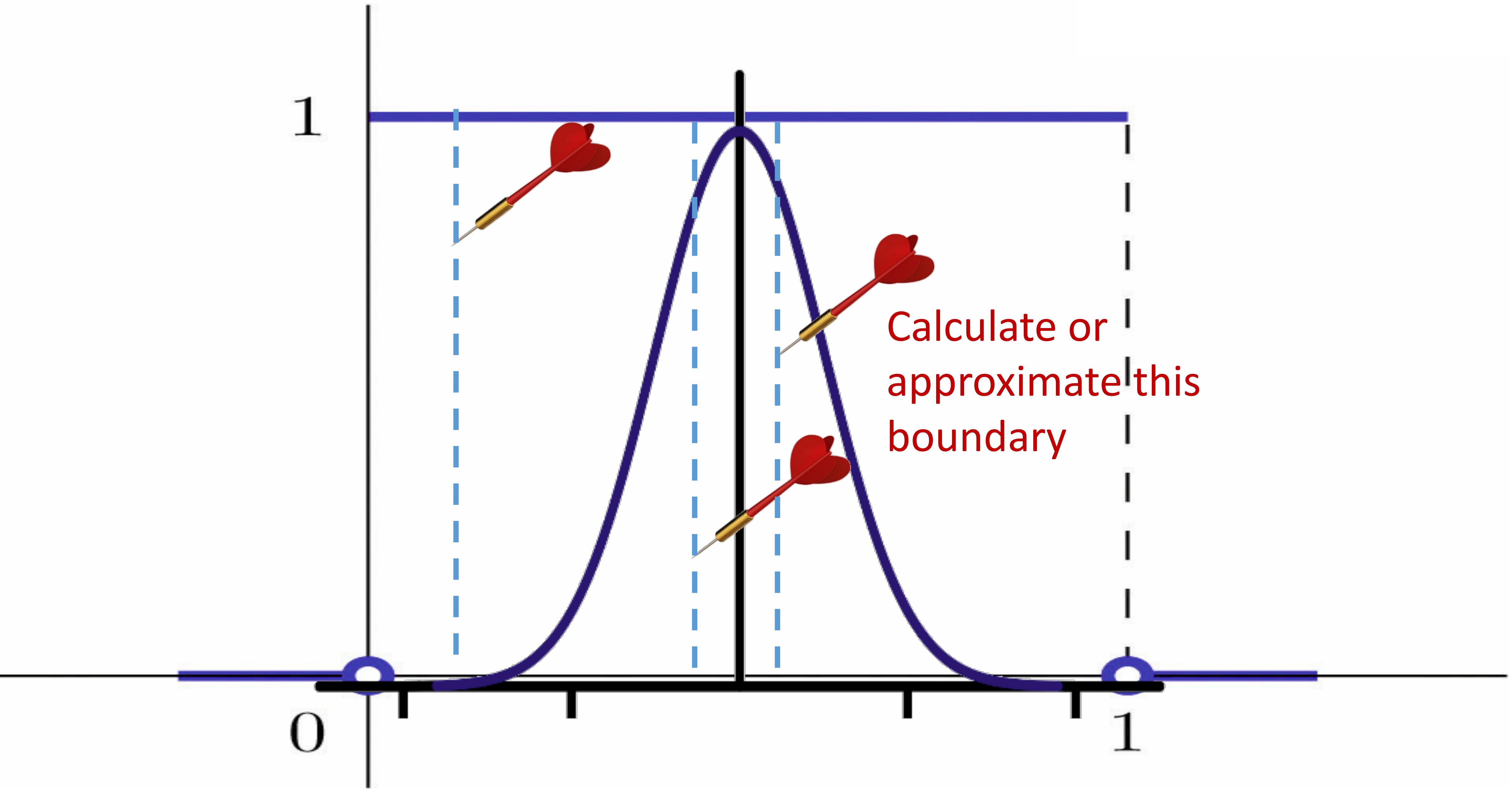
Exercise: use regular expressions to extract the breed from these headers (e.g. the word "breed" and the word after it)

```
>NC_018723.2 Felis catus isolate Cinnamon breed Abyssinian chromosome A1, Felis_catus_8.0, whole genome shotgun sequence
>NC_018724.2 Felis catus isolate Cinnamon breed Abyssinian chromosome A2, Felis_catus_8.0, whole genome shotgun sequence
>NC_018725.2 Felis catus isolate Cinnamon breed Abyssinian chromosome A3, Felis_catus_8.0, whole genome shotgun sequence
```

More on random numbers: random.random() returns a uniformly distributed random value between 0 and 1



What if we want to sample random numbers from another distribution?

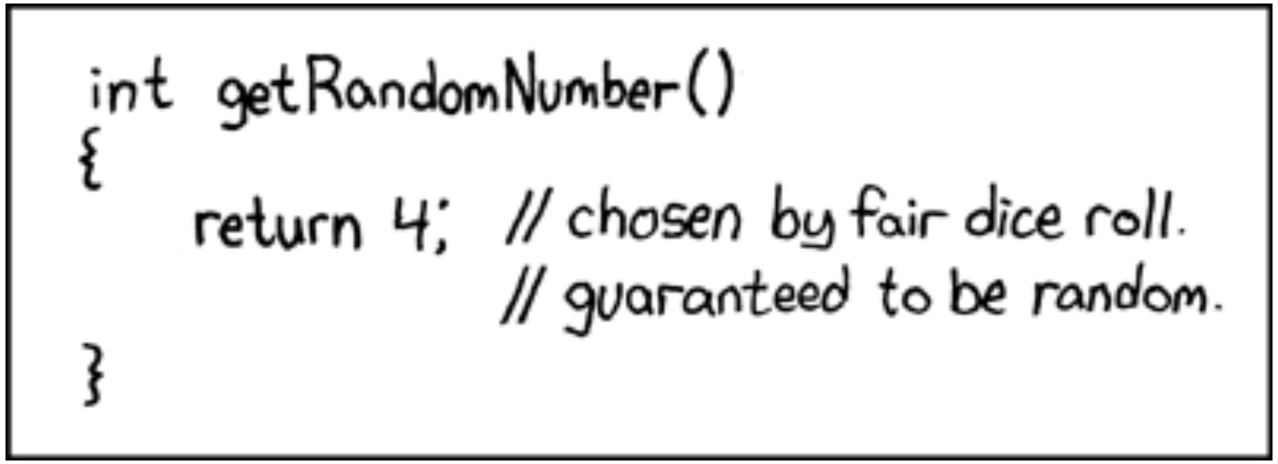

1
0
1
Calculate or approximate this boundary

Python has additional useful random functions

https://docs.python.org/2/library/random.html

```
>>> random.choice(['apple','banana','pear'])
'pear'
>>> random.randint(10,100)
55
>>> random.gauss(0,1) #mean 0, std dev 1
-0.1175
```

What does it mean to generate random numbers anyway?

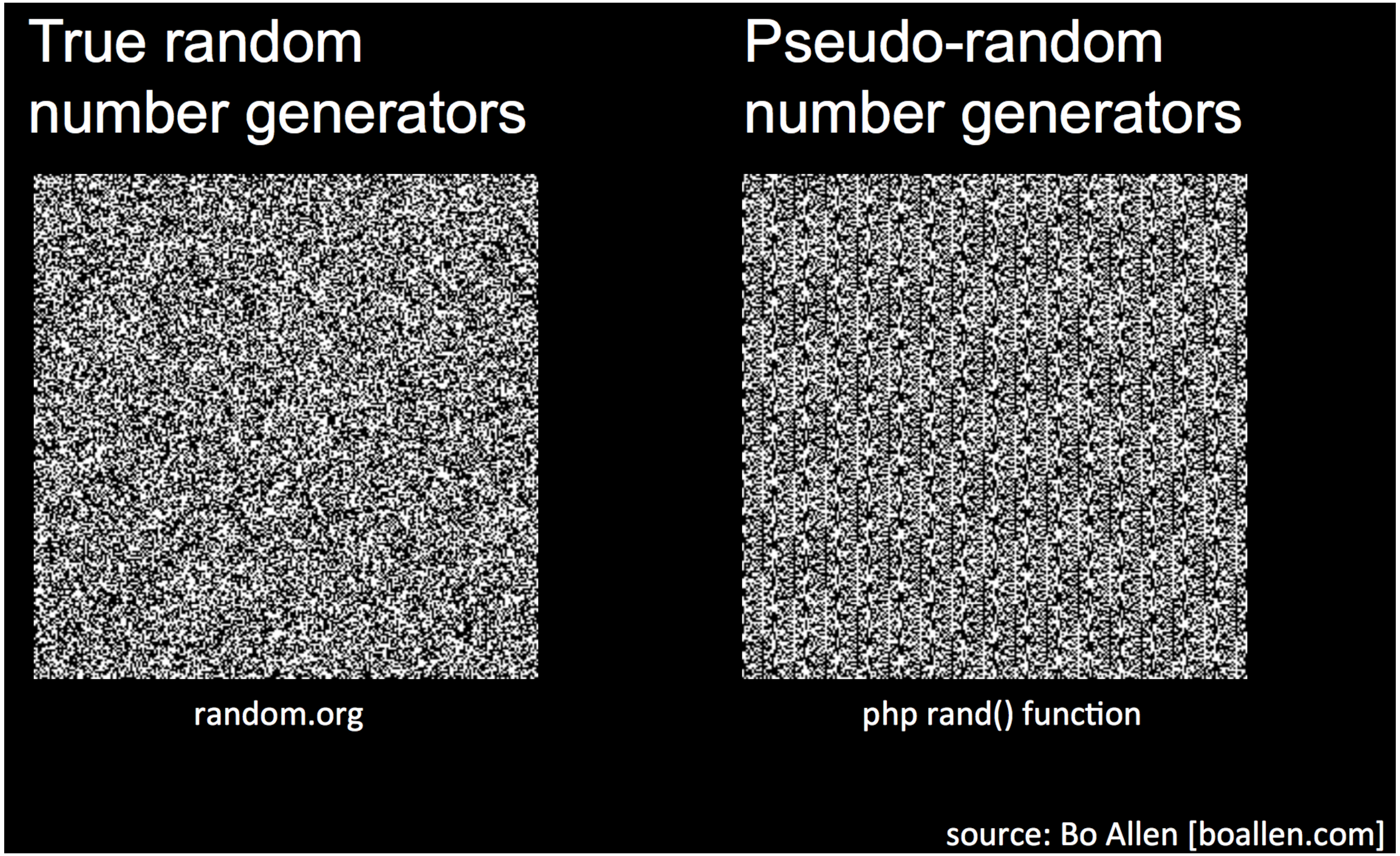


```
int getRandomNumber()
{
    return 4;  // chosen by fair dice roll.
               // guaranteed to be random.
}
```

Computation is deterministic!

Most random number generating algorithms are *pseudo-random*

What are some truly random processes?


True random number generators
Pseudo-random number generators
random.org
php rand() function
source: Bo Allen [boallen.com]

Pseudo-random number generators require a "starting point" called a *seed*

- a seed lets us initialize the random number generator
  - If you know the seed, the sequence of numbers is predictable and fixed
  - If you don't know the seed, the sequence is hopefully unpredictable (but still deterministic)

```
>>> random.seed(number)
```

- You can set the seed to be something unpredictable, like a function of the time at which the code is running

Why might we want predictable "random" numbers?

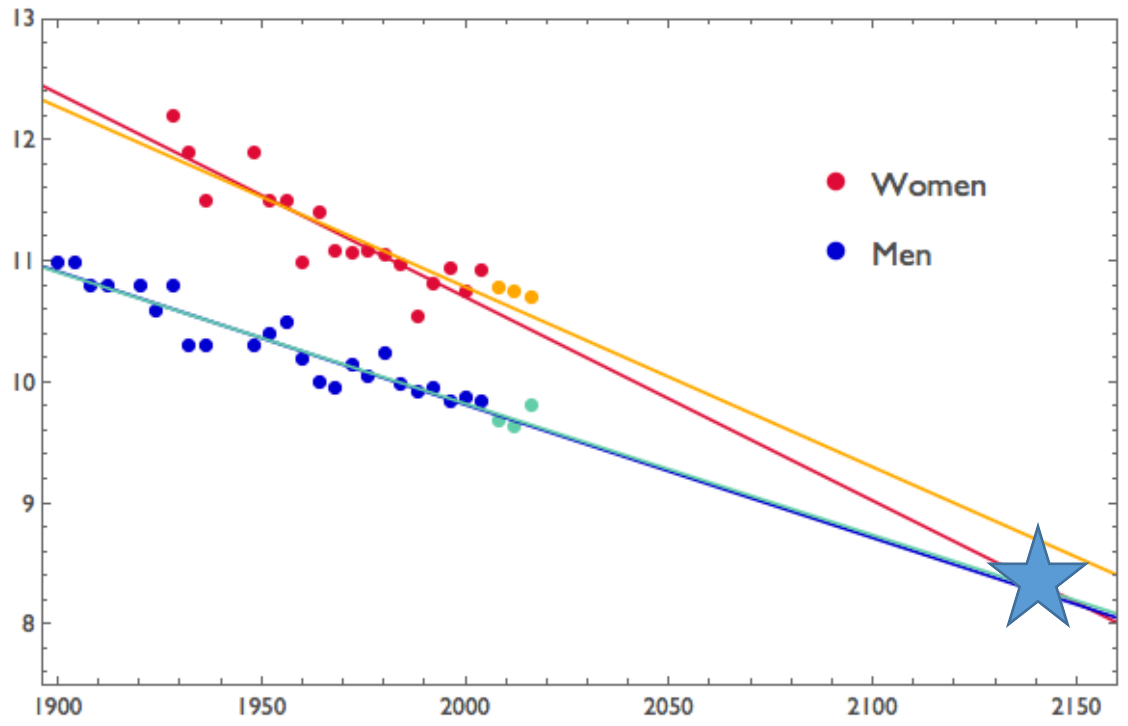
Why might we want unpredictable "random" numbers?

A few things to remember when thinking about machine learning models

- You can build a model of anything, but it doesn't necessarily answer your question
  - Garbage in, garbage out: poorly designed experiments, biased data
    - Batch effects
  - What are the assumptions? Statistical skepticism

- Subtle overfitting
  - Changing model choices based on held-out data
  - Pre-selecting features based on training data and not accounting for it

100m dash Olympic gold medal times
(Tatem *Nature* 2004)



Questions for evaluating classifiers

(read the methods!)

- What and how many features? How were they chosen?
- How many samples in the training dataset? Is it balanced between positive and negative?
- What model was used? Are its assumptions valid?
- Are we evaluating on training, validation, or test data?
- What are the limits of the training and testing data? How generalizable is this model?
  - Example: variant effect predictors only trained and tested on European genetic backgrounds
  - Others?
- What metrics were used for evaluation? What metrics are not shown?
  - http://sphweb.bumc.bu.edu/otlt/mph-modules/bs/bs704_probability/bs704_probability4.html
- …