Simulation

Probability

- Probability allows us to quantify statements about the chance of an event taking place.

For example - Flip a fair coin

1. What's the chance it lands heads?
2. Flip it 4 times, what proportion of heads do you expect?
3. Will you get exactly that proportion?
4. What happens when you flip the coin 1000 times?

4 Flips

- In 4 flips, we can get 0, 1, 2, 3, or 4 Heads and so the proportion of Heads can be: 0, 0.25, 0.5, 0.75, or 1
- We expect the proportion to be 0.5
- But, a proportion of 0.25 is quite likely:

There are 16 possible ways for 4 tosses to land, e.g.
HHHH, HHHT, HHTH, …

Each is equally likely, so the chance of any particular sequence of Hs and Ts is 1/16
So chance of 0.25 proportion is 4/16
HTTT, THTT, TTHT, TTTH

4 Flips

- We can think of the proportion of Heads in 4 flips as a statistic because it summarizes data

- Notice that it is a random quantity – it takes on 5 possible values, each with some probability

value	0.00	0.25	0.50	0.75	1.00
chance	1/16	4/16	6/16	4/16	1/16

1,000 Flips

- When we flip the coin 1,000 times, we can get a many different possible proportions of Heads, i.e. 0, 0.001, 0.002, 0.003, …, 0.998, 0.999, 1.000

- It's highly unlikely that we would get 0 for the proportion – how unlikely?

- What does the distribution of the proportion of heads in 1000 flips look like?

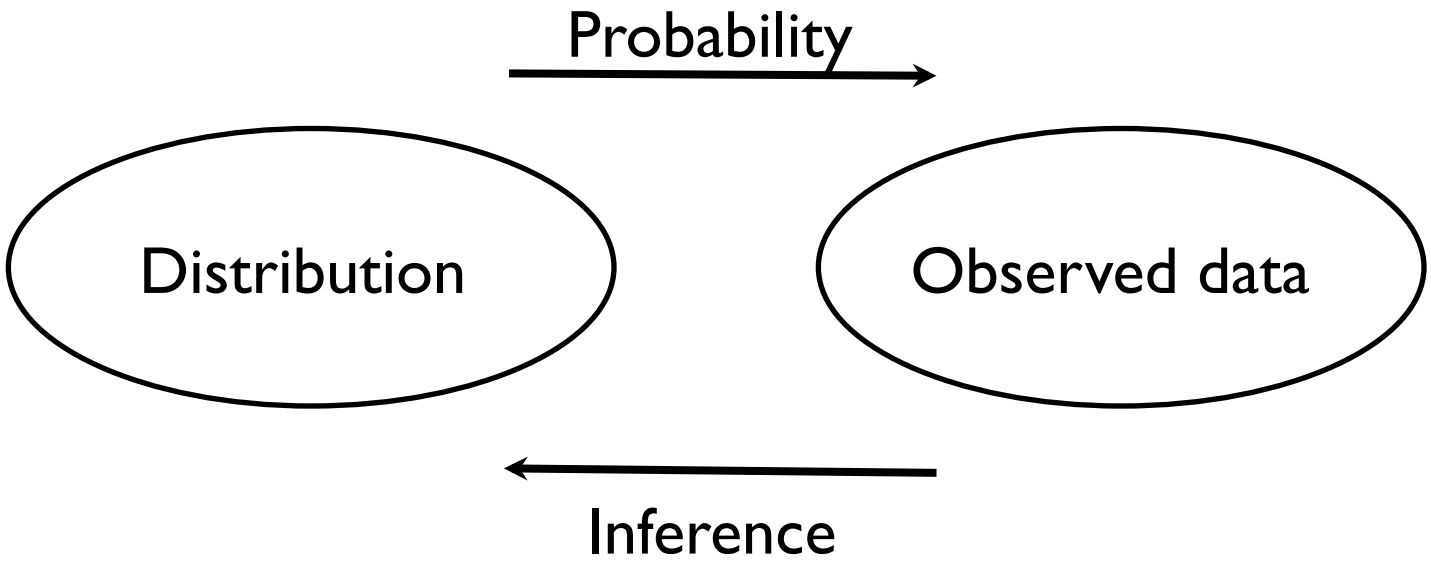
1,000 Flips

- With some advanced math tools, we can figure this out.
- But we can also get a good idea using a simulation.
- In our simulation we will assume that the chance of Heads is 0.5 and find out what the possible values for the proportion of heads in 1,000 flips looks like
- If we were to carry out an experiment with a coin and get a particular proportion, say 0.37, then we could use this simulation study to help us understand the results of our experiment.

Let's Generalize

Before we consider the role that simulation can play in helping us understand statistics, let's take a step back and think about the big picture.

We can think of probability theory as complimentary to statistical inference.



A *statistic* is often just a function of a random sample, for example the sample mean, the 95th quantile, or the sample proportion.

Statistics are often used as *estimators* of quantities of interest about the distribution, called *parameters*. Statistics are random variables (since they depend on the sample); parameters are not.

In simple cases, we can study the *sampling distribution* of the statistic analytically. For example, we can prove that under mild conditions the distribution of the sample proportion is close to normal for large sample sizes.

In more complicated cases, we turn to simulation.

The main idea in a simulation study is to replace the mathematical expression for the distribution with a *sample* from that distribution.

In our example: $X_1, X_2, \ldots, X_n$ are independent observations from the same distribution.

The distribution has center (mean/expected value) μ and spread (standard deviation) σ

We are interested in the distribution of $\overline{X}$

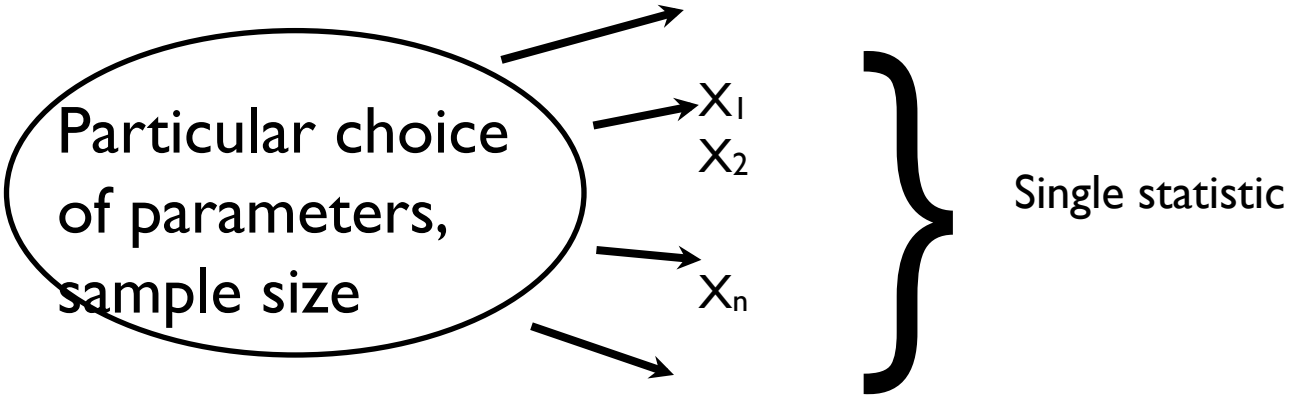
So we take many samples of size n, and study the behavior of the sample averages

We use the sample to *estimate* features of the distribution, such as the behavior of various statistics under repeated sampling from the distribution.

This set of techniques, sometimes called Monte Carlo methods, is very powerful. Statisticians routinely use it to evaluate complicated methods for which exact mathematical results are difficult or impossible to obtain.

The downside: whereas mathematical results are symbolic, in terms of arbitrary parameters and sample size, in a simulation we must specify particular values.

A *single experiment within a simulation* looks like this:



To approximate the *sampling distribution* of the statistic, we repeat the whole experiment B times. The larger B is, the better our approximation will tend to be.

Steps in carrying out a simulation study:

1. Specify what makes up an individual experiment: sample size, distributions, parameters, statistic of interest.
2. Write an expression or function to carry out an individual experiment and return the statistic.
3. Determine what inputs, if any, to vary (e.g. different sample sizes or parameters).
4. For each combination of inputs, repeat the experiment B times, providing B samples of the statistic.
5. For each combination of inputs, summarize the *empirical distribution* of the statistic of interest.
6. State and/or plot the results. (Sometimes go back to 3.)

Example: Carry out a simulation study of the median when sampling from the normal distribution. How does it vary with the sample size and with the standard deviation of the normal distribution?

Useful Random Number Generators

```
sample(x, size, replace = FALSE,
       prob = NULL)
```

Think of an urn with tickets, each ticket marked with a value. Mix up the tickets and draw one at a time from the urn

- x = vector with one element for each ticket, values correspond to what is written on the ticket.
- size = number of draws to take from the urn
- replace = replace the ticket between draws or not.
- prob = set of weights for the elements in x (an element might represent more than one ticket)

Useful Random Number Generators

Standard Probability Distributions:

runif(n, min = 0, max = 1) – sample from the uniform distribution on the interval (0, 1).

So the chance the value drawn is:

between 0 and 1/3 has chance 1/3;
between 1/3 and 1/2 has chance 1/6;
between 9/10 and 1 has chance 1/10

The min and max allow you to change the interval from which to sample, e.g. min = 100, max = 150 will produce random values between 100 and 150

Useful Random Number Generators

Standard Probability Distributions:

rnorm(n, mean = 0, sd = 1) – sample from the normal distribution with center = mean and spread = sd

rbinom(n, size, prob), - sample from the binomial distribution with number of trials = size and chance of success = prob

Other distributions: rexp(), rpois(), rt(), rf() – each has arguments for parameter values relevant to the distribution … See ?Distributions for more information

How does R generate random numbers?

Actually, it doesn't

R uses a **pseudo random number generator**:

- It starts with a **seed** and an **algorithm** (i.e. a function)
- The seed is plugged into the algorithm and a number is returned
- That number is then plugged into the algorithm and the next number is created

The algorithms are such that the numbers produced behave/look like random values

Simple Congruential Generator

The congruential method uses modular arithmetic to generate "random" numbers.

From inputs a and b and an initial value, x_0 , the first "random number" is generated as follows:

$$x_1 = a * x_0 \bmod b$$

and subsequent numbers are generated recursively,

$$x_{(n+1)} = a * x_n \bmod b$$

We call x_0 the **seed**

Congruential a = 3, b = 64

Seed = 17

3 * 17 mod 64 = 51 mod 64 = 51

3 * 51 mod 64 = 153 mod 64 = 25

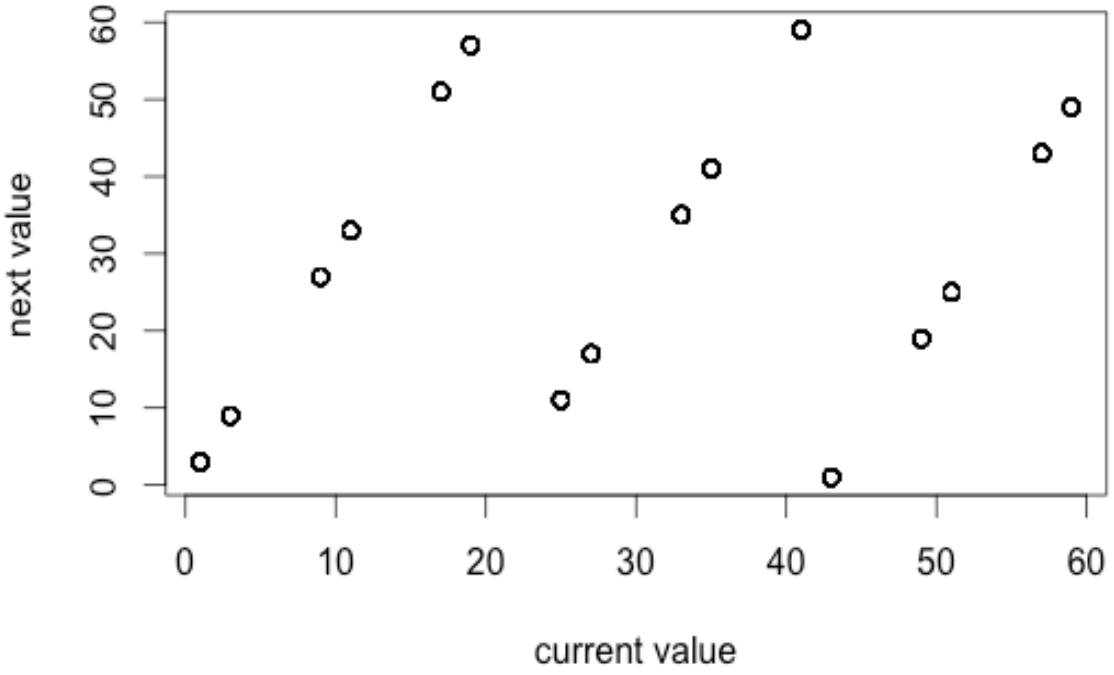
3 * 25 mode 64 = 75 mod 64 = 11

And so on. The first 20 "random" numbers are

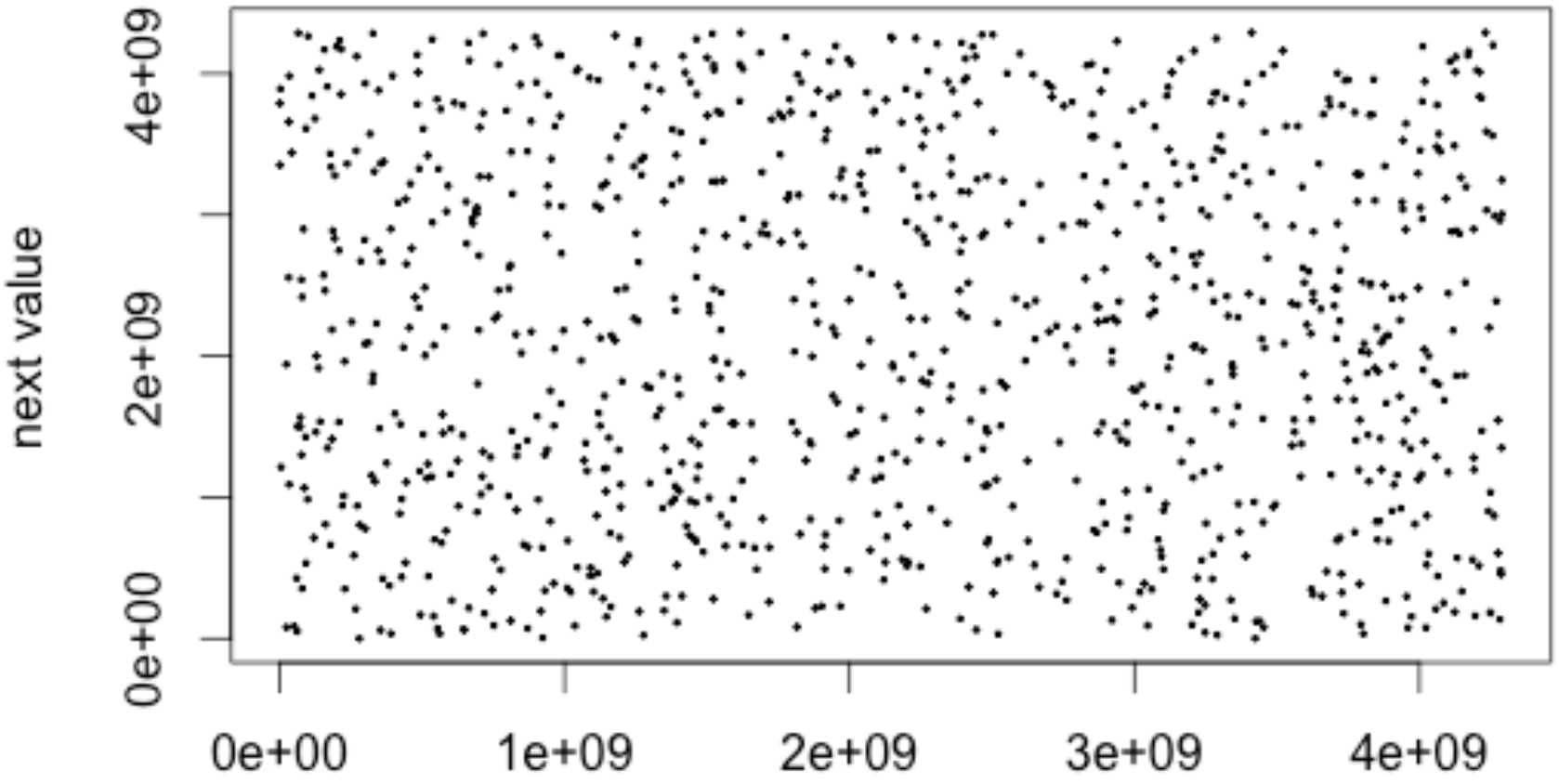
```
51 25 11 33 35 41 59 49 19 57 43  1  3  9 27
17 51 25 11 33
```

Generate 1000 values

```
plot(x3b64[1:(n-1)], x3b64[2:n],
 xlab = "current value”.vlab = "next value")
```



```
cong(n, a = 69069, b = 2^32)
```



The Seed

There is one big advantage to pseudo-random number generators:

You can reproduce your simulation results by controlling the seed:

`set.seed()` allows you to do this:

When researchers publish results from simulation studies, they typically include the random number generator and the seed that was used so that others can verify/replicate their results

The Seed

```
> set.seed(69069)
```
Set the seed for the RNG

```
> runif(3)
[1] 0.1648855 0.9564664 0.3345479
```
Call the uniform RNG

```
> runif(3)
[1] 0.01109596 0.18654873 0.94657805
```
Call the uniform RNG again

```
> set.seed(69069)
> runif(3)
[1] 0.1648855 0.9564664 0.3345479
```
Set the seed back to 69069

The for statement

Looping is the repeated evaluation of a statement or block of statements.

Much of what is handled using loops in other languages can be more efficiently handled in R using vectorized calculations or one of the apply mechanisms.

However, certain algorithms, such as those requiring recursion, can only be handled by loops.

There are two main looping constructs in R: `for` and `while`.

For loops

A *for loop* repeats a statement or block of statements a predefined number of times.

The syntax in R is

```
for ( name in vector ){
  statement
}
```

For each element in `vector`, the variable `name` is set to the value of that element and `statement` is evaluated.

`vector` often contains integers, but can be any valid type.

While loops

A *while loop* repeats a statement or block of statements for as many times as a particular condition is `TRUE`.

The syntax in R is

```
while (condition){
  statement
}
```

`condition` is evaluated, and if it is `TRUE`, statement is evaluated. This process continues until condition evaluates to `FALSE`.

Exercise:

The expression

```
Sample(1:0, size = 1, prob = c(p, 1-p))
```

simulates a random coin flip, where the coin has probability `p` of coming up heads, represented by a `1`.

Write a function that simulates flipping a coin until a fixed number of heads are obtained. It should take the probability `p` and the total number of heads `total` and return the trial on which the final head was obtained. This produces a single sample from the *negative binomial distribution*.

The break statement causes a loop to exit. This is particularly useful with while loops, which, if we're not careful, might loop indefinitely (or until we kill R).

```
# Simulate number of tosses to get 10 Heads

max.iter = 1000
x = 0
steps = 0
while(x < 10){
  x = x + sample(c(0, 1), 1)
  steps = steps + 1
  if(steps == max.iter){
    warning("Maximum iteration reached")
    break
  }
}
```

Biham-Middleton-Levine traffic model

A Simulation Study

http://www.mathinstitutes.org/nuggets/traffic_gridlock.html

Question(s)

- Traffic flow in a large city grid
- Is there a largest traffic density that permits free flow?
- Is there a density above which gridlock is inevitable?
- In 1992 Biham, Middleton and Levine introduced a simplified model for the study of these questions, called the BML model.

BML Model

- Each intersection of a square grid of streets contains either a red car, a blue car, or an empty space.
- At each odd-numbered time step, all blue cars simultaneously attempt to move one unit North;
- A car succeeds if there is already an empty space for it to move into.
- At each even-numbered time step, the red cars attempt to move East in the same way.

BML Model

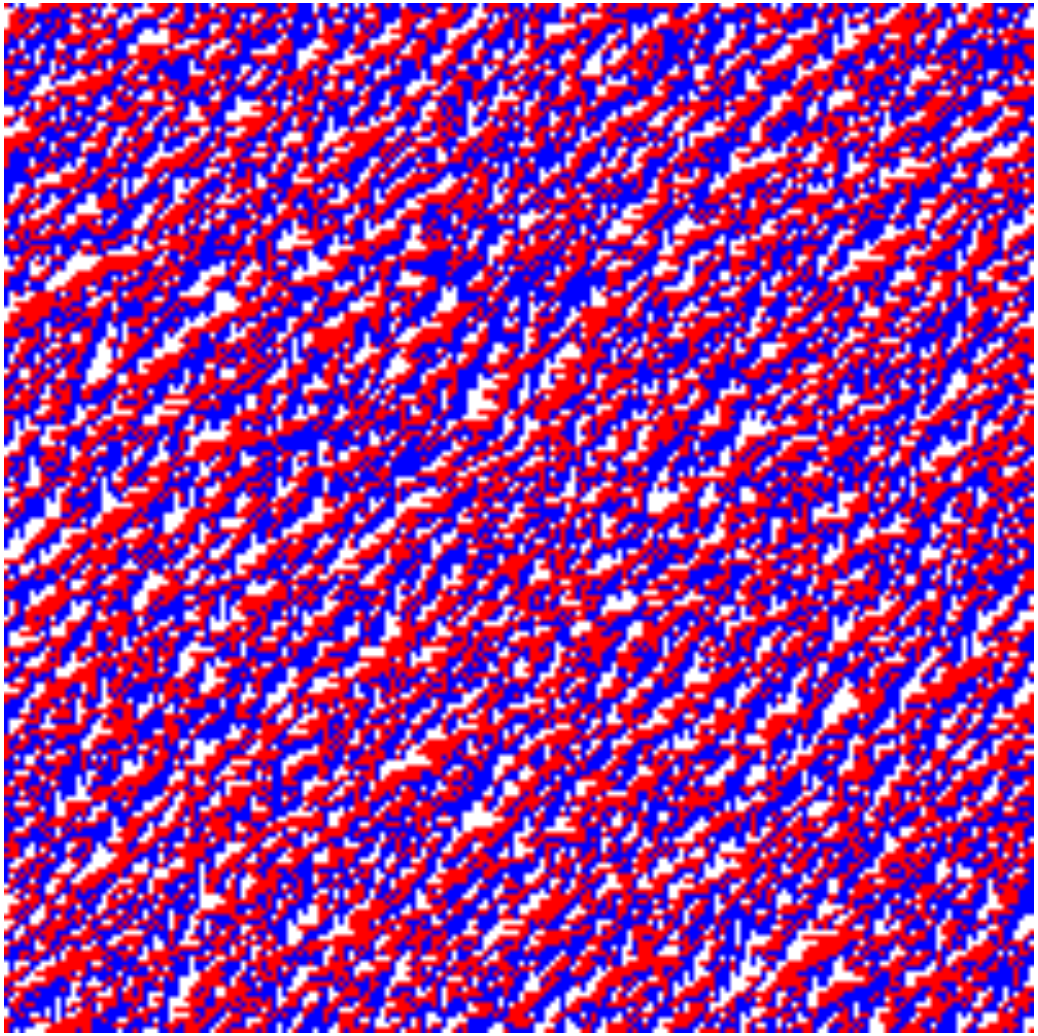
- A blue car that falls off the North edge of the grid reappears in the same position at the South edge;
- Similarly red cars falling off the East edge reappear on the West edge.
- Initially, cars are distributed at random: each intersection is independently assigned a car with probability p, or an empty space with probability 1 - p.
- Each is car is independently equally likely to be red or blue.

BML Model

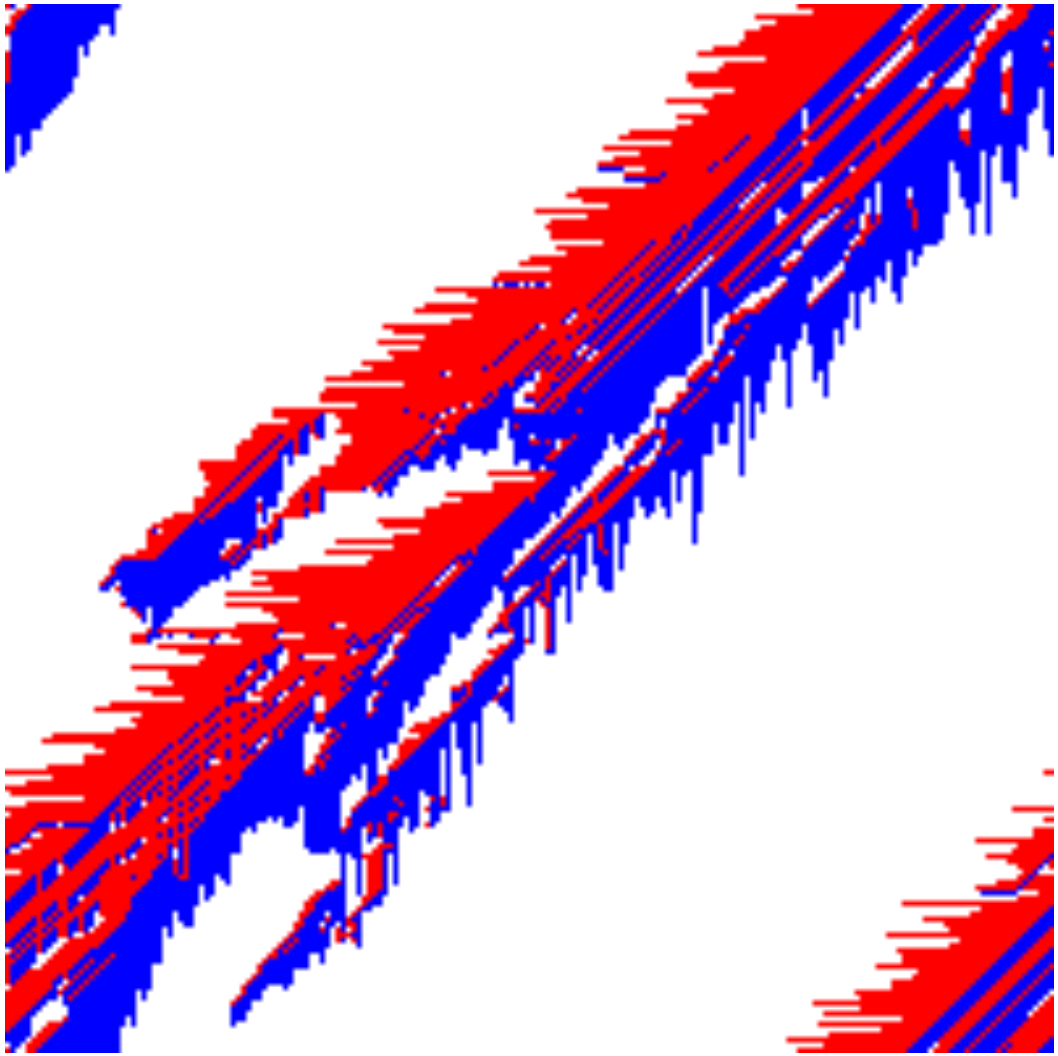
Is there a density above which gridlock is inevitable?

- BML is perhaps the simplest system exhibiting phase transitions and self organization
- General belief: the system exhibits a sharp phase transition from free flowing to fully jammed, depending on **p** – the initial density of cars.

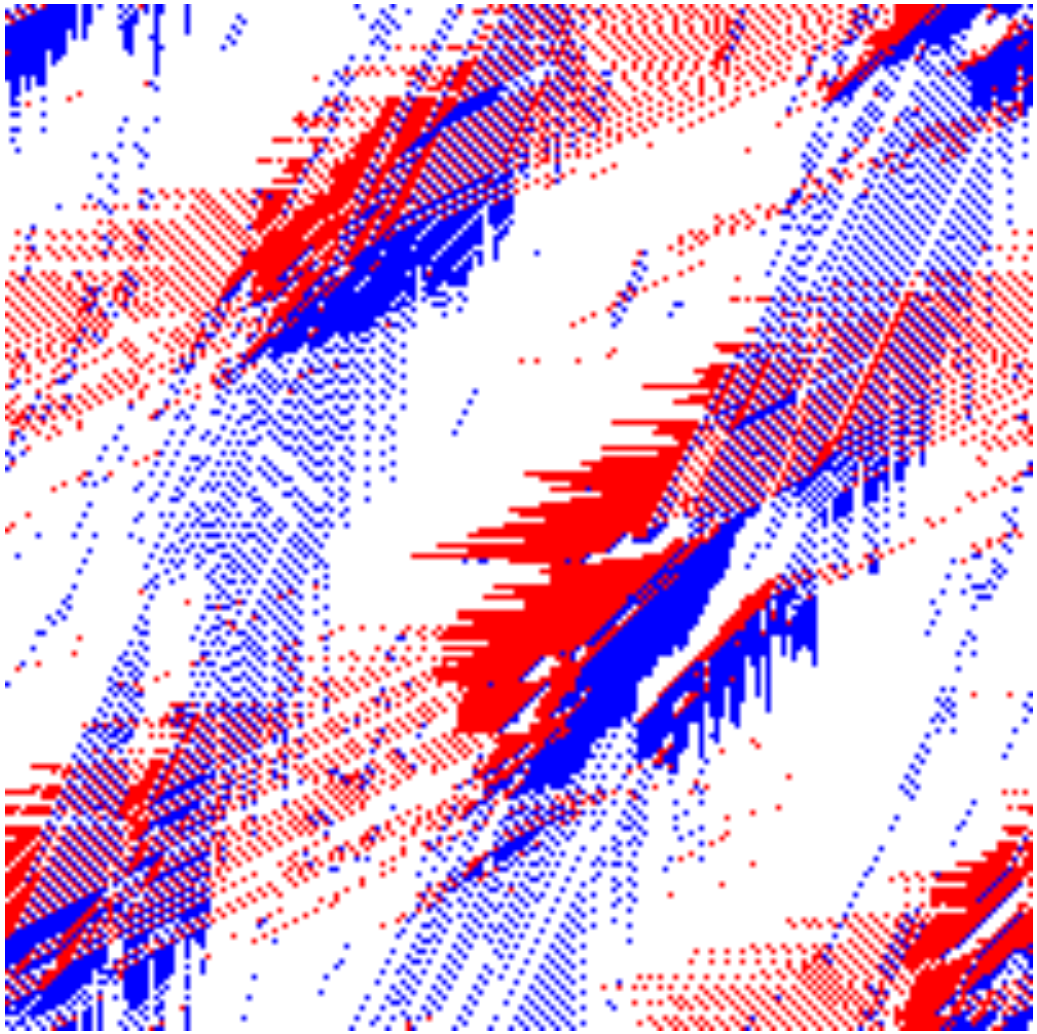
BML p=0.80



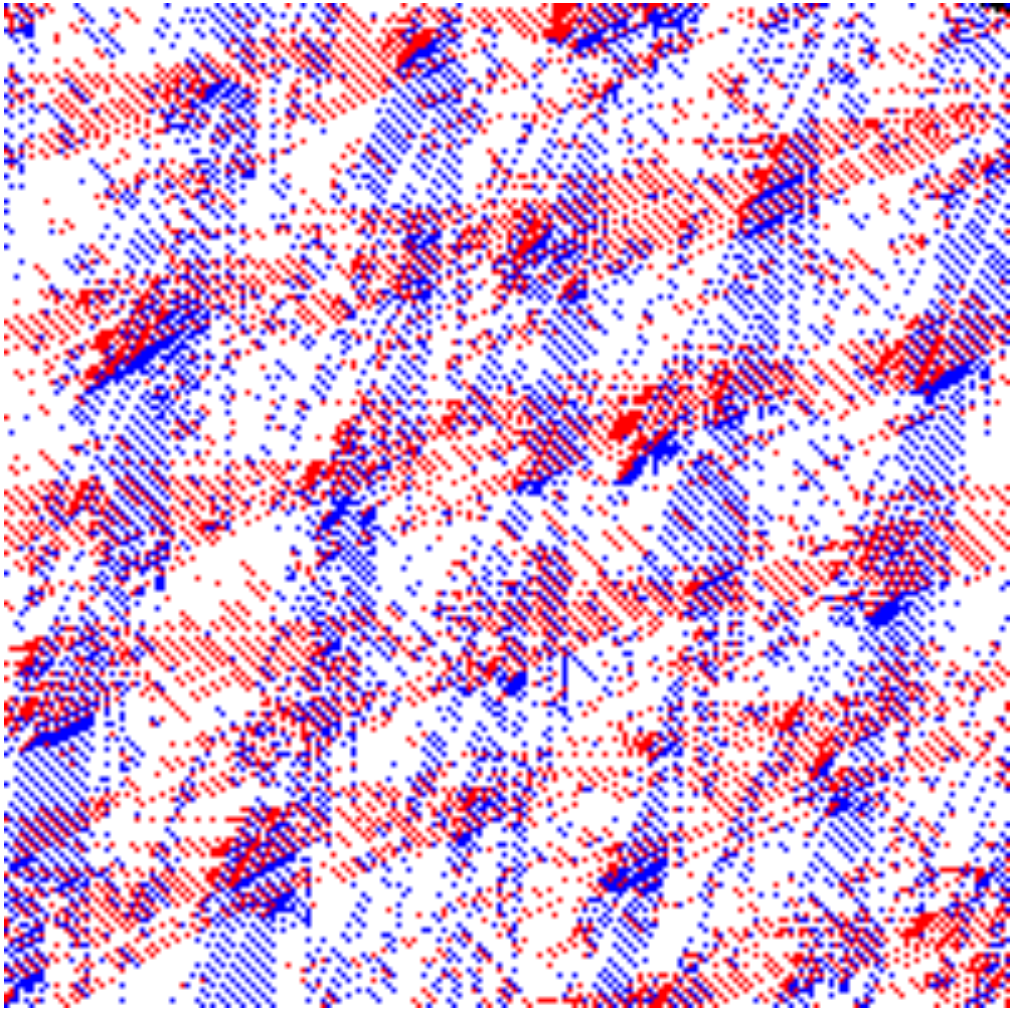
BML p=0.34



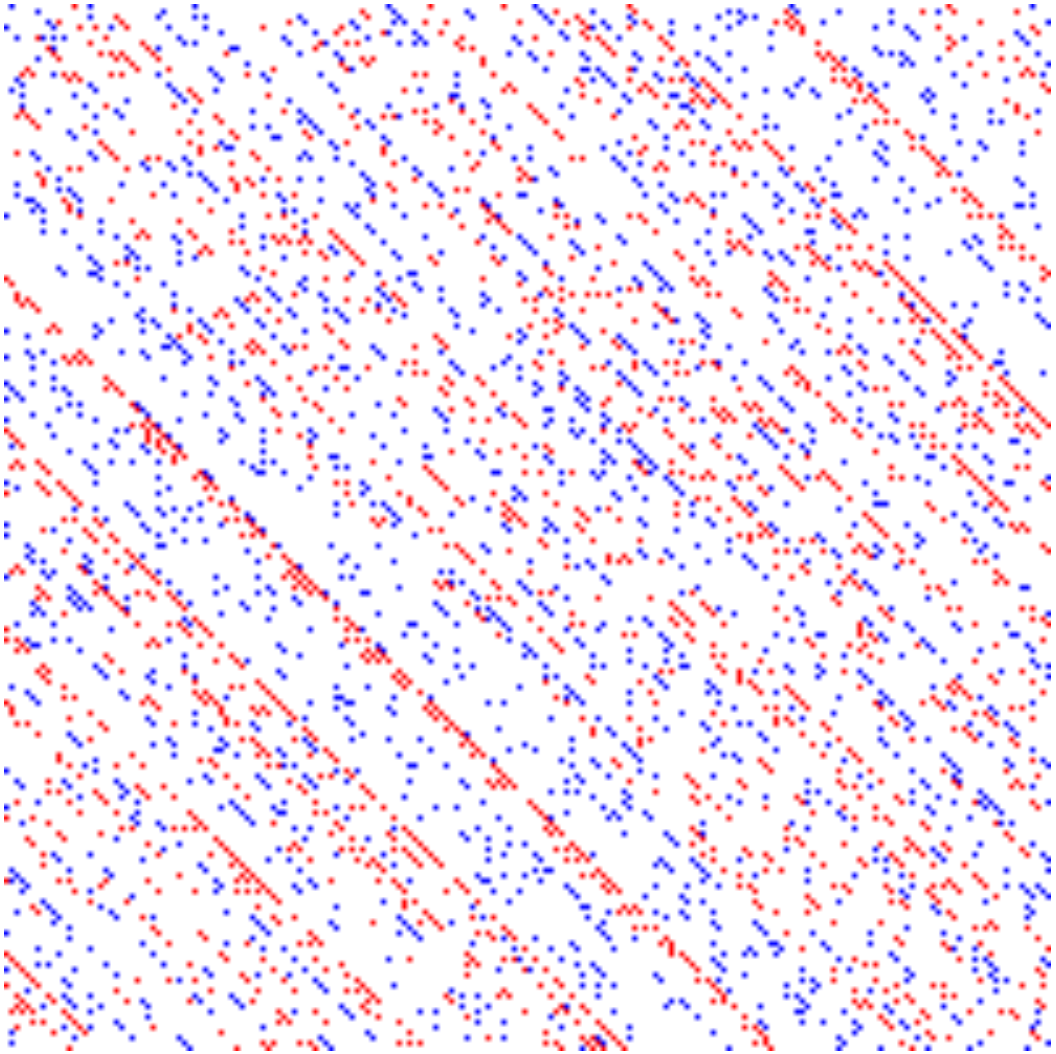
BML p=0.32



BML p=0.32



BML p=0.10



BML Model

- These images show outcomes after 20,000 steps.

- When p = 80%, traffic becomes jammed, and no car can move at all.
- For low densities (e.g. 10%), after a while traffic is completely free flowing, and no car ever has to wait at all.

BML Model

- The model appears to also exhibit large-scale organization:
- When p is 34% there is a single jam spanning the entire grid,
- In the 30% picture the cars have arranged themselves into wide diagonal bands that avoid each other.
- In the two 32% pictures, all cars move some of the time and wait some of the time, and this is achieved by semi-regular geometric patterns of jams feeding into each other.

BML Model

- Video at

www.math.ucdavis.edu/~njlinesch/BML/

Example: Carry out a simulation study of the median when sampling from the normal distribution. How does it vary with the sample size and with the standard deviation of the normal distribution?

Experiment:

- Generate `n` random normal values from a Normal(0, `s^2`)
- Take the median of these `n` values

```
n = 27
s = 3
median(rnorm(n = n, sd = s))
```

Repeat Experiment:

- Repeat the experiment `B` times
- Examine the distribution of the `B` medians

```
B = 1000
sampleMs = replicate(1000,
       median(rnorm(n = n, sd = s))
mean(sampleMs)
sd(sampleMs)
hist(sampleMs)
```

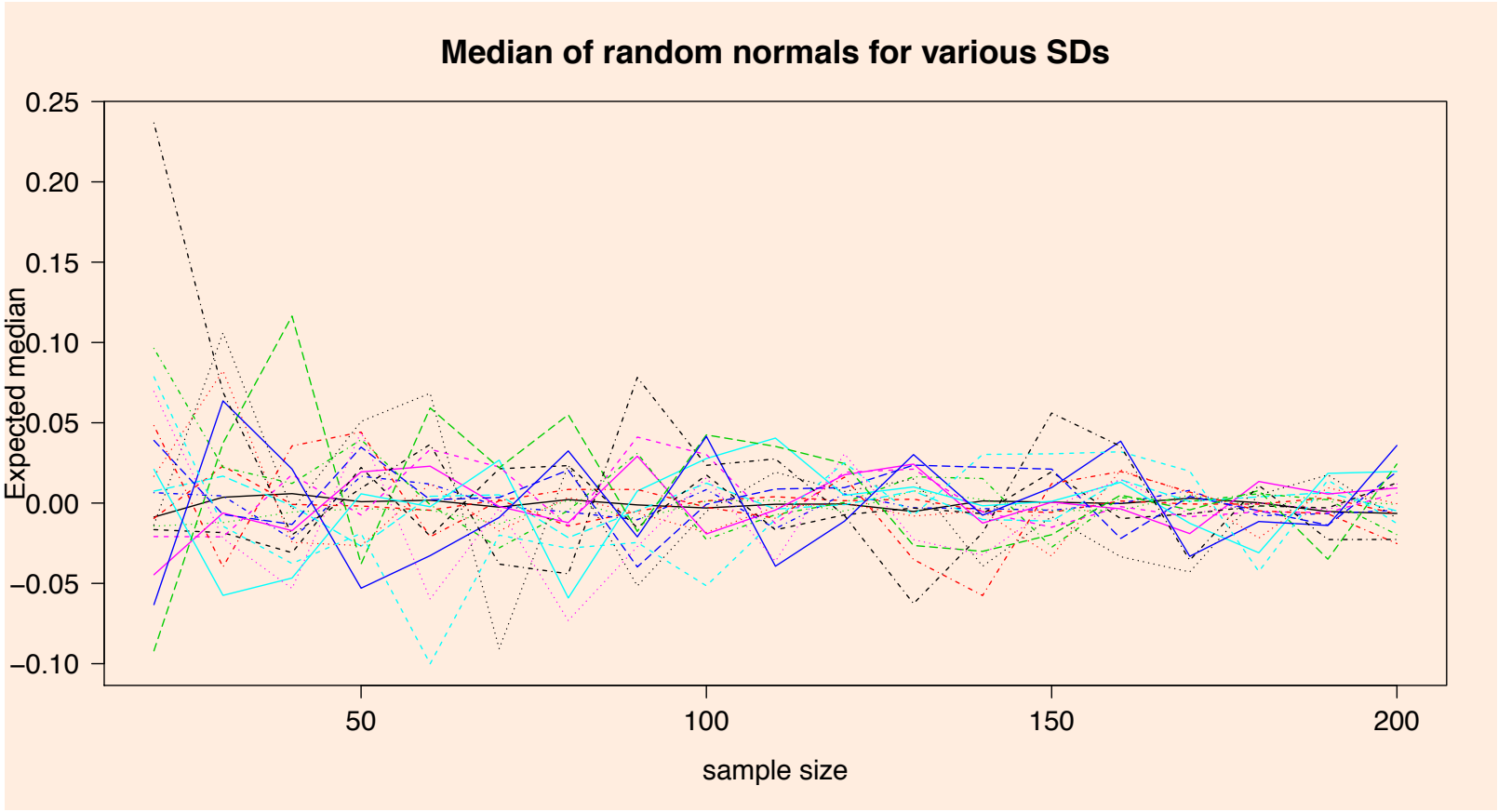
Repeat Simulation

- Repeat the simulation for different values of `n` and `s`
- Compare/Examine the behavior for these different values

```
ns = seq(20, 200, by = 10)
ss = seq(1, 10, by = 0.5)
```

Repeat Simulation

```
ns = seq(20, 200, by = 10)
ss = seq(1, 10, by = 0.5)
samples = matrix(nrow = length(ns),
                 ncol = length(ss))
for (i in 1:length(ns)) {
  for (j in 1:length(ss)) {
   samples[i , j] =
    mean(replicate(1000,
           median(rnorm(n = ns[i],
                        sd = ss[j]))))
  }
}
```


Median of random normals for various SDs
Expected median
0.25
0.20
0.15
0.10
0.05
0.00
−0.05
−0.10
50
100
150
200
sample size

The Life Cycle of Data

Data Science Pipeline

- Data science is an evolving field
- Based on elements of computer science, statistics, engineering, and information science.
- But it is a science! What does this mean?

Science vs Non-Science

- What makes certain activities “science” vs other activities?

- Goal: discover facts about our world.

- But, like in the statistical paradigm, we never know the Truth, and we never know whether we are right.

- The best we can hope for is to become more of less certain of a conjecture.

Feynman (1956)

"… it is imperative in science to doubt; it is absolutely necessary, for progress in science, to have uncertainty as a fundamental part of your inner nature. To make progress in understanding we must remain modest and allow that we do not know. Nothing is certain or proved beyond all doubt. You investigate for curiosity, because it is *unknown*, not because you know the answer. And as you develop more information in the sciences, it is not that you are finding out the truth, but that you are finding out that this or that is more or less likely.

That is, if we investigate further, we find that the statements of science are not of what is true and what is not true, but statements of what is known to different degrees of certainty: "It is very much more likely that so and so is true than that it is not true;" or "such and such is almost certain but there is still a little bit of doubt;" or – at the other extreme – "well, we really don't know." Every one of the concepts of science is on a scale graduated somewhere between, but at neither end of, absolute falsity or absolute truth."

Merton's Scientific Norms (1942)

Communalism: scientific results are the common property of the community.

Universalism: all scientists can contribute to science regardless of race, nationality, culture, or gender.

Disinterestedness: act for the benefit of a common scientific enterprise, rather than for personal gain.

Originality: scientific claims contribute something new

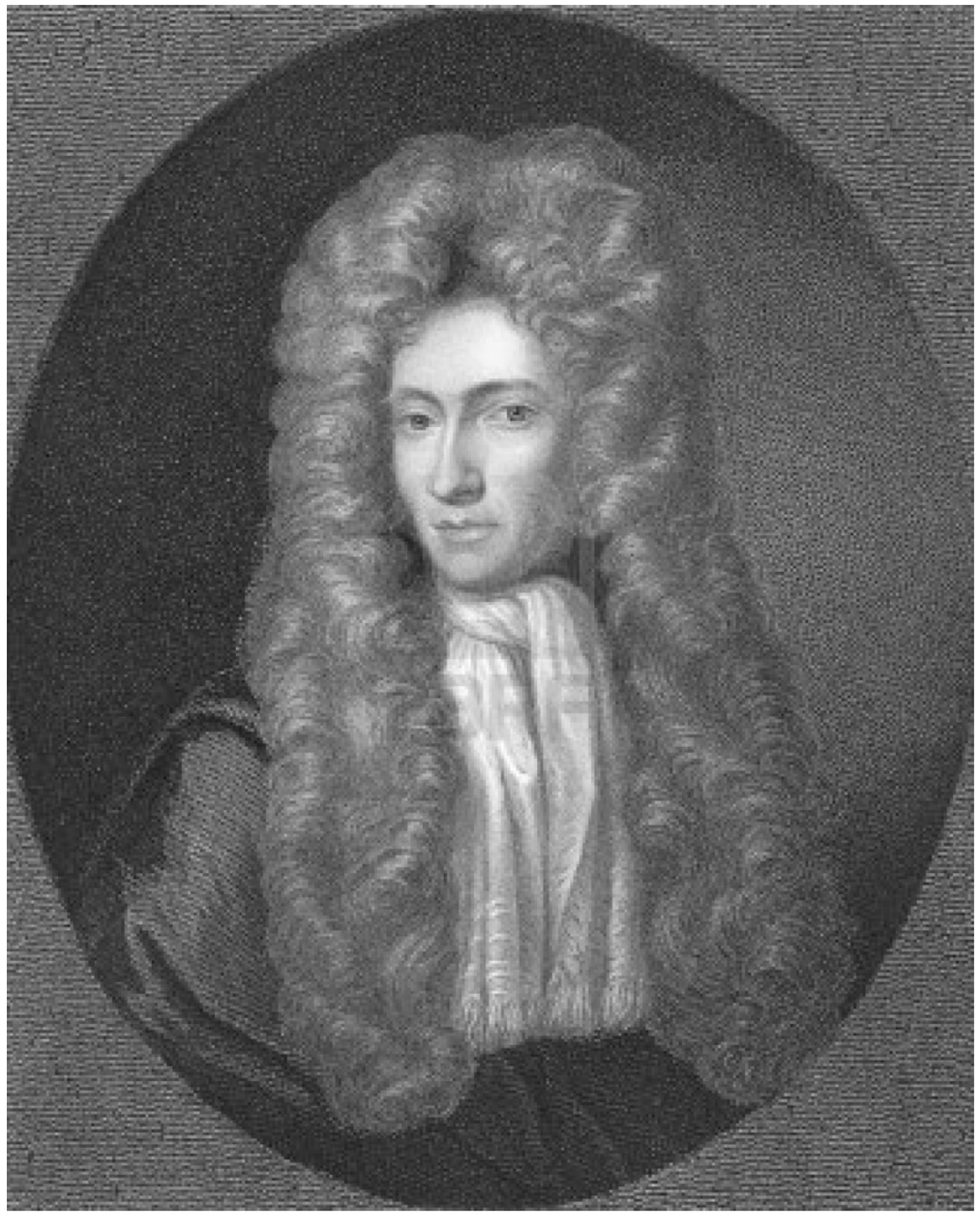
Skepticism: scientific claims must be exposed to critical scrutiny before being accepted.

Skepticism -> Reproducibility

- Skepticism requires that the claim can be independently verified,

- This in turn requires transparency in the communication of the research process.

- Instantiated by Robert Boyle and the Transactions of the Royal Society in the 1660's.



Back to Data Science

- Traditional data discovery paradigm:
    1. researcher develops a hypothesis,
    2. designs the experiment to test the hypothesis,
    3. collects data,
    4. tests hypothesis.

- Data-driven discovery paradigm:
    1. researcher finds data (often very large and disparate),
    2. explores and investigates data,
    3. generates a hypothesis (or prediction),
    4. tests hypothesis (or prediction).

Data Science and Skepticism

- Traditional paradigm: carefully record all relevant details of the experimental and hypothesis testing process. Make these details available when publishing the finding.
    - for example, the methods section in a scientific publication, or the lab notebook.
- Data-driven discovery paradigm: the same!

Data Science and Skepticism

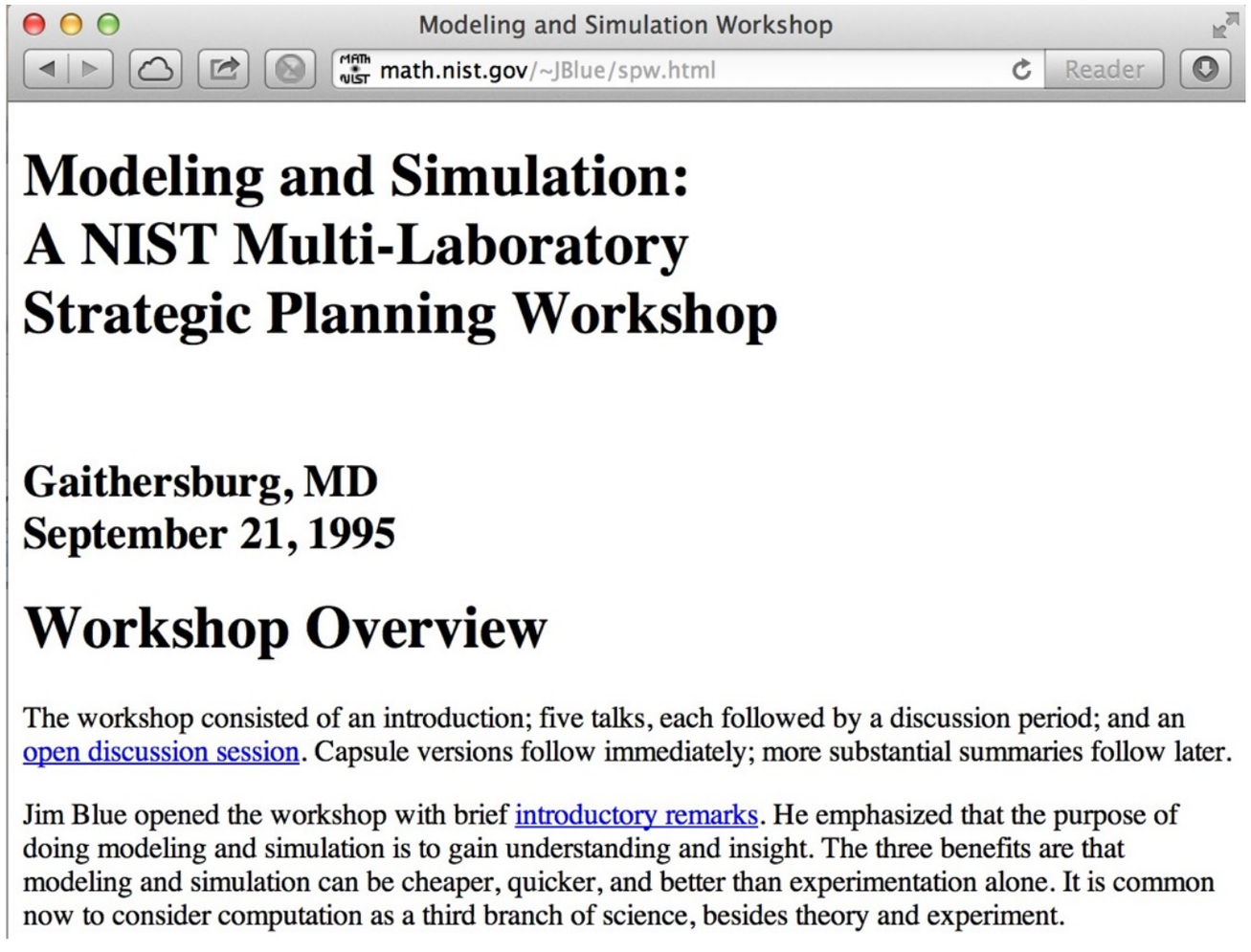
Traditionally two branches to the scientific method:

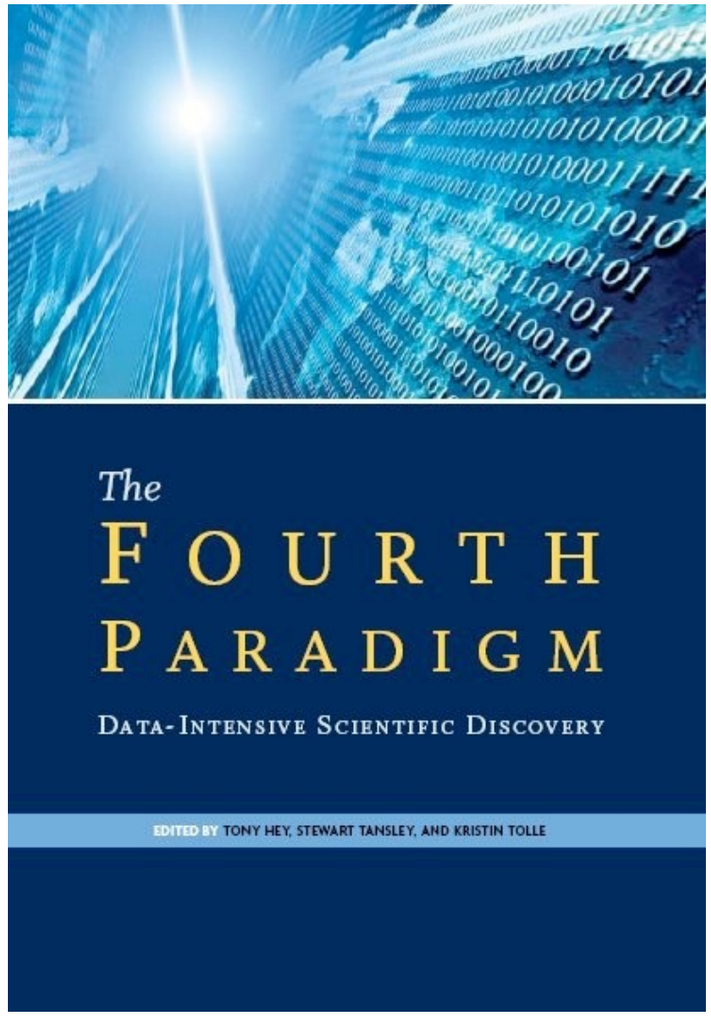
- Branch 1 (deductive): mathematics, formal logic,
- Branch 2 (empirical): statistical analysis of controlled experiments.

Now, new branches due to technological changes?

- Branch 3,4? (computational): large scale simulations / data driven computational science.

Some Evidence..





"It is common now to consider computation as a third branch of science, besides theory and experiment."

"This book is about a new, fourth paradigm for science based on data-intensive computing."

The Ubiquity of Error

The central motivation for the scientific method is to root out error:

- Deductive branch: the well-defined concept of the proof,
- Empirical branch: the machinery of hypothesis testing, appropriate statistical methods, structured communication of methods and protocols.

Claim: Computation presents only a *potential* third/fourth branch of the scientific method (Donoho, Stodden, et al. 2009), until the development of comparable standards.

Really Reproducible Research

"Really Reproducible Research" (1992) inspired by Stanford Professor Jon Claerbout:

> "The idea is: An article about computational science in a scientific publication is not the scholarship itself, it is merely advertising of the scholarship. The actual scholarship is the complete ... set of instructions [and data] which generated the figures." David Donoho, 1998

Note the difference between: reproducing the computational steps and, replicating the experiments independently including data collection and software implementation. (Both required)

So what should data scientists do?

- Record all details of your experiment, like a lab notebook.
- Make these details available with the publication of results.
- Typically this means a data scientist will be sharing their code and data, along with their results.
- Code and data should be re-usable: other researchers should be able to generate your figures and tables on their own. The “Life Cycle” of data.

Anything else?

- Yes! Later in the class we will talk about statistical procedures and techniques that encourage reproducible statistical inferences. (avoiding overfitting, avoiding multiple comparisons for example.)

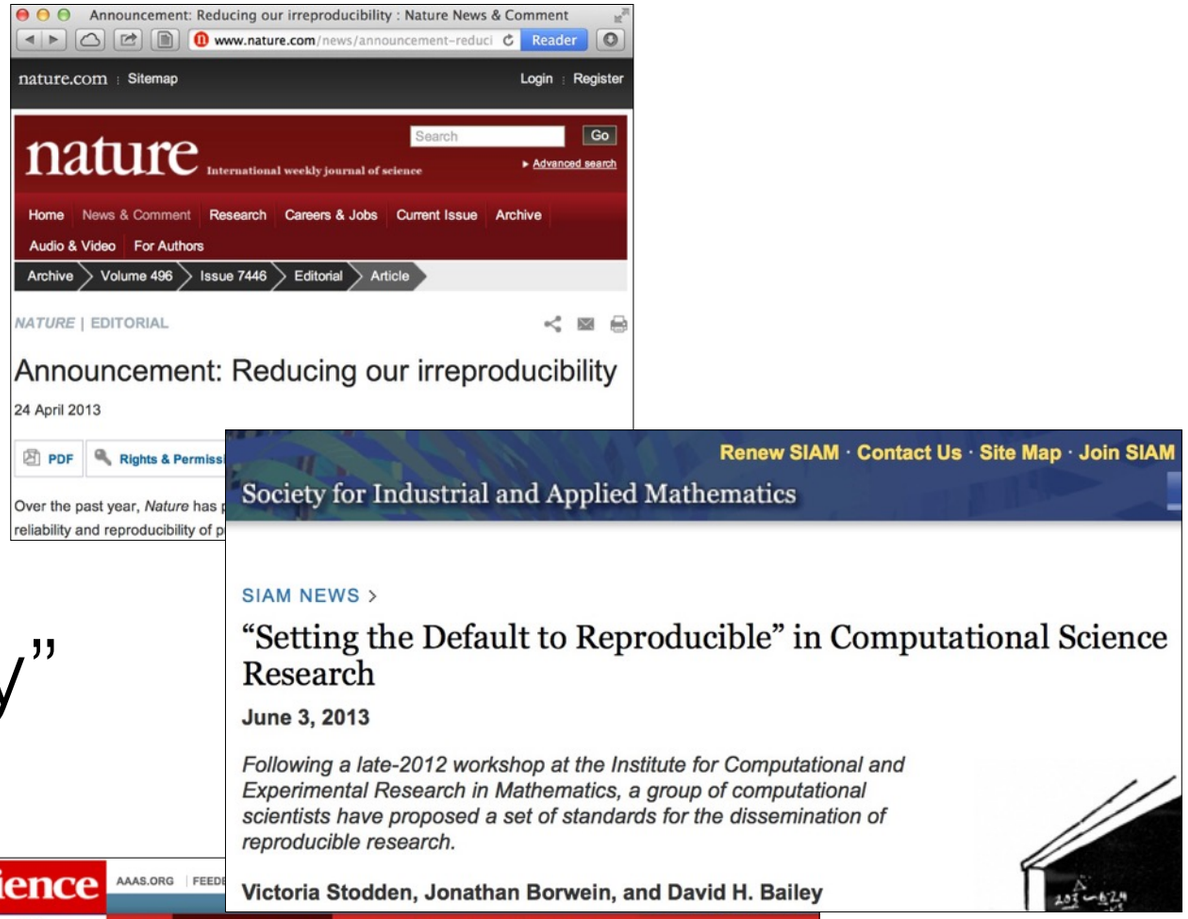
- That leads us to discuss different types of reproducibility in science: *empirical*, *computational*, and *statistical*.

Parsing Reproducibility

"Empirical Reproducibility"

"Computational Reproducibility"

"Statistical Reproducibility"



V. Stodden, IMS Bulletin (2013)

Empirical Reproducibility

Cell Reports
Commentary

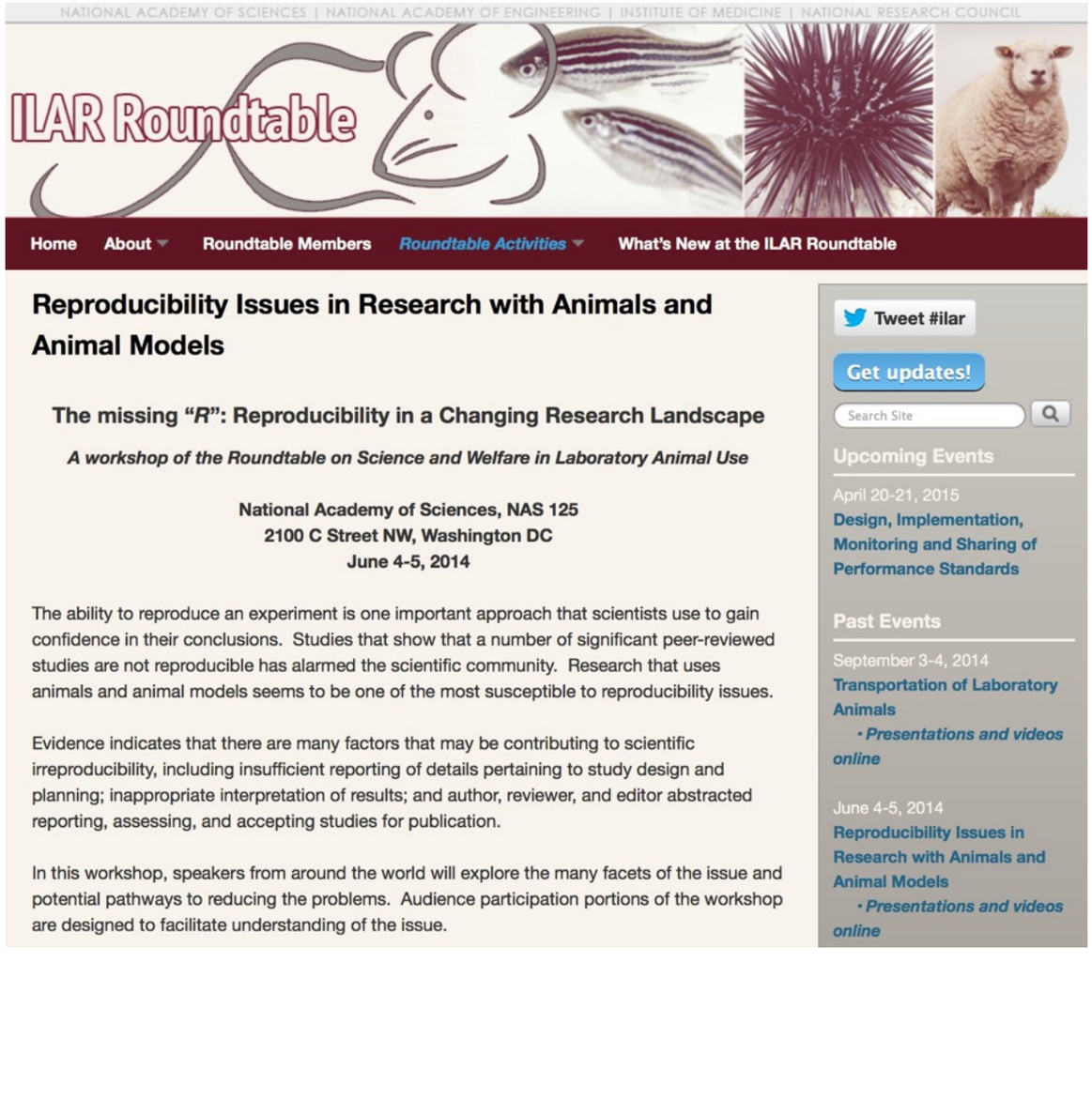
Sorting Out the FACS: A Devil in the Details

William C. Hines,[1,5,*] Ying Su,[2,3,4,5,*] Irene Kuhn,[1] Kornelia Polyak,[2,3,4,5] and Mina J. Bissell[1,5]
[1]Life Sciences Division, Lawrence Berkeley National Laboratory, Mailstop 977R225A, 1 Cyclotron Road, Berkeley, CA 94720, USA
[2]Department of Medical Oncology, Dana-Farber Cancer Institute, Boston, MA 02215, USA
[3]Department of Medicine, Brigham and Women's Hospital, Boston, MA 02115, USA
[4]Department of Medicine, Harvard Medical School, Boston, MA 02115, USA
[5]These authors contributed equally to this work
*Correspondence: chines@lbl.gov (W.C.H.), ying_su@dfci.harvard.edu (Y.S.)
http://dx.doi.org/10.1016/j.celrep.2014.02.021

The reproduction of results is the cornerstone of science; yet, at times, reproducing the results of others can be a difficult challenge. Our two laboratories, one on the East and the other on the West Coast of the United States, decided to collaborate on a problem of mutual interest—namely, the heterogeneity of the human breast. Despite using seemingly identical methods, reagents, and specimens, our two laboratories quite reproducibly were unable to replicate each other's fluorescence-activated cell sorting (FACS) profiles of primary breast cells. Frustration of studying cells close to their context in vivo makes the exercise even more challenging.

Paired with in situ characterizations, FACS has emerged as the technology most suitable for distinguishing diversity among different cell populations in the mammary gland. Flow instruments have evolved from being able to detect only a few parameters to those now capable of measuring up to—and beyond—an astonishing 50 individual markers per cell (Cheung and Utz, 2011). As with any exponential increase in data complexity, breast reduction mammoplasties. Molecular analysis of separated fractions was to be performed in Boston (K.P.'s laboratory, Dana-Farber Cancer Institute, Harvard Medical School), whereas functional analysis of separated cell populations grown in 3D matrices was to take place in Berkeley (M.J.B.'s laboratory, Lawrence Berkeley National Lab, University of California, Berkeley). Both our laboratories have decades of experience and established protocols for isolating cells from primary normal breast tissues as well as the capabilities required for

Cell PRESS
Open ACCESS



Computational Reproducibility

Digital Aspects of Research:

1. Big Data / Data Driven Discovery: high dimensional data, p >> n,

2. Computational Power: simulation of the complete evolution of a physical system, systematically varying parameters,

3. Deep intellectual contributions now encoded only in software.

What we share as data scientists must reflect the changing nature of research, and include code, data, workflows.

CSHL Keynote; Dr. Lior Pachter, UC Berkeley

The software contains “ideas that enable biology...”
Stories from the Supplement, 2013

Statistical Reproducibility

- False discovery, “p-hacking,” file drawer problem, overuse and mis-use of p-values, lack of multiple testing adjustments.

- Low power, poor experimental design, nonrandom sampling,

- Data preparation, treatment of outliers, re-combination of datasets, insufficient reporting/tracking practices,

- inappropriate tests or models, model misspecification,

- Model robustness to parameter changes and data perturbations,

- Investigator bias toward previous findings; conflicts of interest.

We will return to these topics..

Final Thoughts

- Prepare data and code *as you work* for sharing when you make your results available.

- Share these objects at the same time as publication, not just when/if someone asks (in a persistent repository, such as The DataVerse Network).

- Even if no one ever uses them, you will do better science if you share them, and it will make it possible for others to verify and build on your work.