McCormick

Northwestern Engineering

Why Good Simulations Go Bad

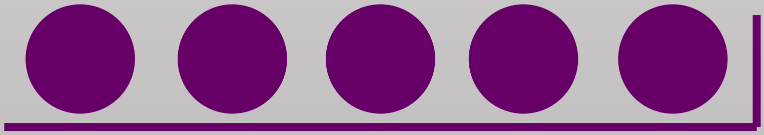
Barry L. Nelson

Department of Industrial Engineering & Management Sciences

7th Simulation Workshop
1-2 April 2014

Formative experiences

- 1977: Dog chasing runner
  - System of differential equations integrated numerically through time.
- 1980: USAF HEART Project
  - Real system, real data, real decisions, real mistakes.
- 1983: M/M/1 queue animation
  - Visualization (programmed on a Commodore Vic-20).



McCormick
Northwestern Engineering

I am a believer in simulation…

- Never has simulation been more used or accepted.
  - Software is cool, pretty and intuitive.
- Emphasis on “analytics” will increase the use of simulation, and slant use more toward decision-support than design.
- Simulations will be integrated into other tools as well as being standalone.
  - Simulations will build themselves from data in enterprise management systems.
- …but I also worry.

McCormick
Northwestern Engineering

Why <u>good</u> simulations go bad

- Simulation is "Risky Business"

- The Ghost of R. A. Fisher

- Simulators don't clean up their messes

- Everything I told you is wrong

The 4 disclaimers

1. This is **not** a research talk.
   - Research tends to emphasize exceptions.
   - This talk is about what is most common.
2. The opinions expressed here are my own and cannot be blamed on the Workshop organizers.
   - They are, however, well-reasoned and insightful.
3. Yes, the talk is a little bit preachy; sorry about that.
4. I am carefully **not** plugging anybody's software.

Risky business: Handball on steroids

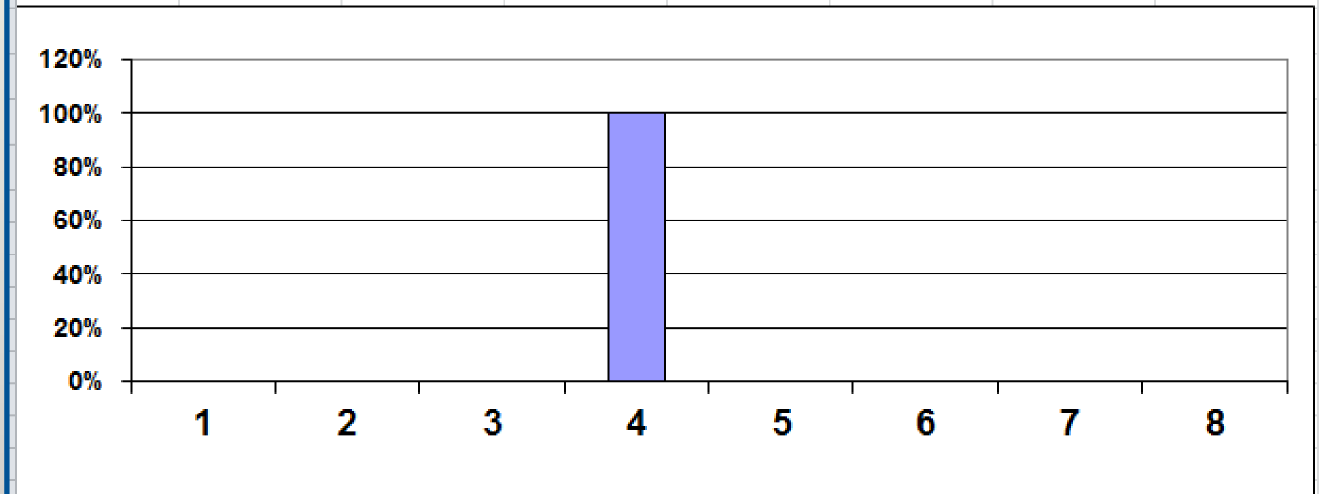
- Jai Alai is a handball-like game on which there is pari-mutuel betting.
- 8 players compete, first to 7 points wins.
- Players play in order, 2 at a time, and hold court if they win.
- 1 point/win for the first 7 games, 2 points/win after that.

Stolen with thanks from *Calculated Bets: Computers, Gambling, and Mathematical Modeling to Win* by Steven Skiena

Does starting position matter? A simulation answer

1. Assume all players are equally good, so Probability{win point} = ½, like a coin flip.
2. Write an algorithm in a computer language that represents the rules of Jai Alai.
3. Let pseudorandom numbers stand in for the coin.
4. Run the algorithm for many matches and record the number of wins by position.

0	0	0	1	0	0	0	0	Wins
0%	0%	0%	100%	0%	0%	0%	0%	Percentage
0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	Error (95% CI)
Player 1	Player 2	Player 3	Player 4	Player 5	Player 6	Player 7	Player 8	



Number of Games

1

Number Played

1

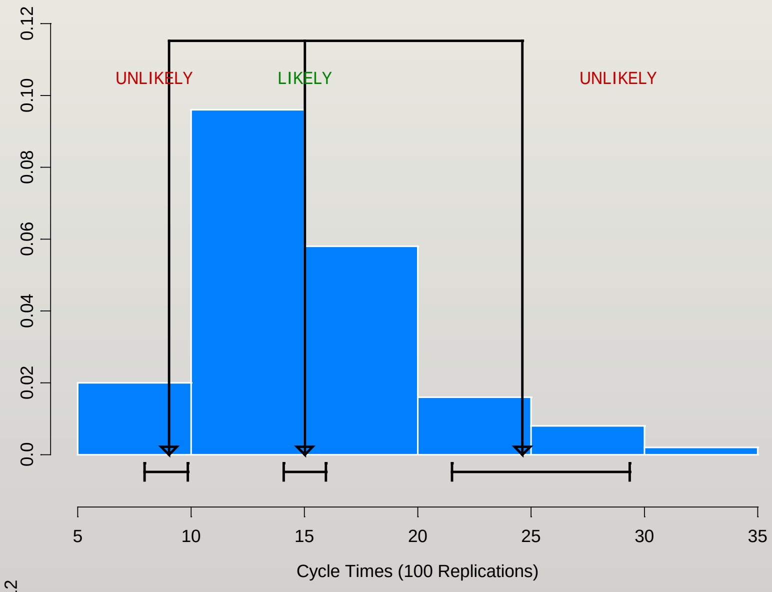
Play Jai Alai

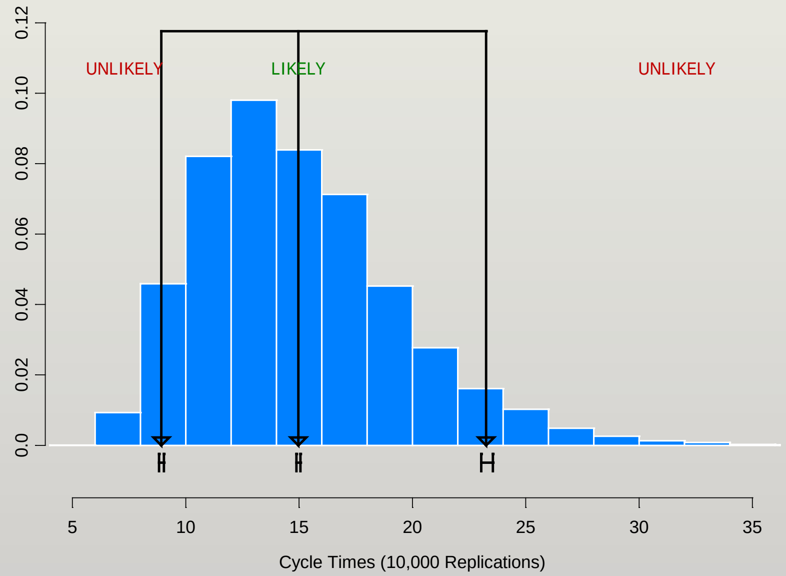
Jai Alai Simulation

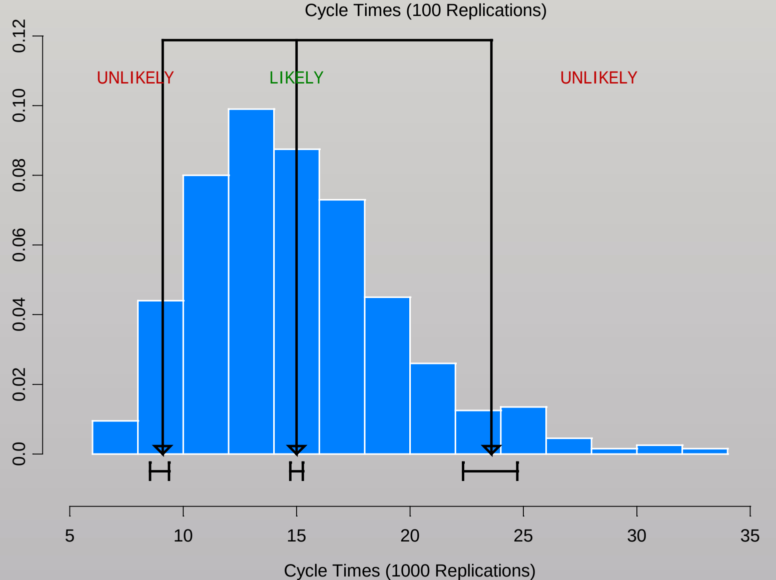
This is useful information, but…

- No one (I hope) thinks that because we estimated these probabilities very precisely that we can guarantee a winning pick.
- No matter how much simulation we do, there is still **RISK** on any actual bet.
- Yet in simulation we have focused obsessively on measuring/controlling/displaying **ERROR** in estimating the mean rather than conveying the **RISK** in making a decision.

Eye chart examples: Order fill time

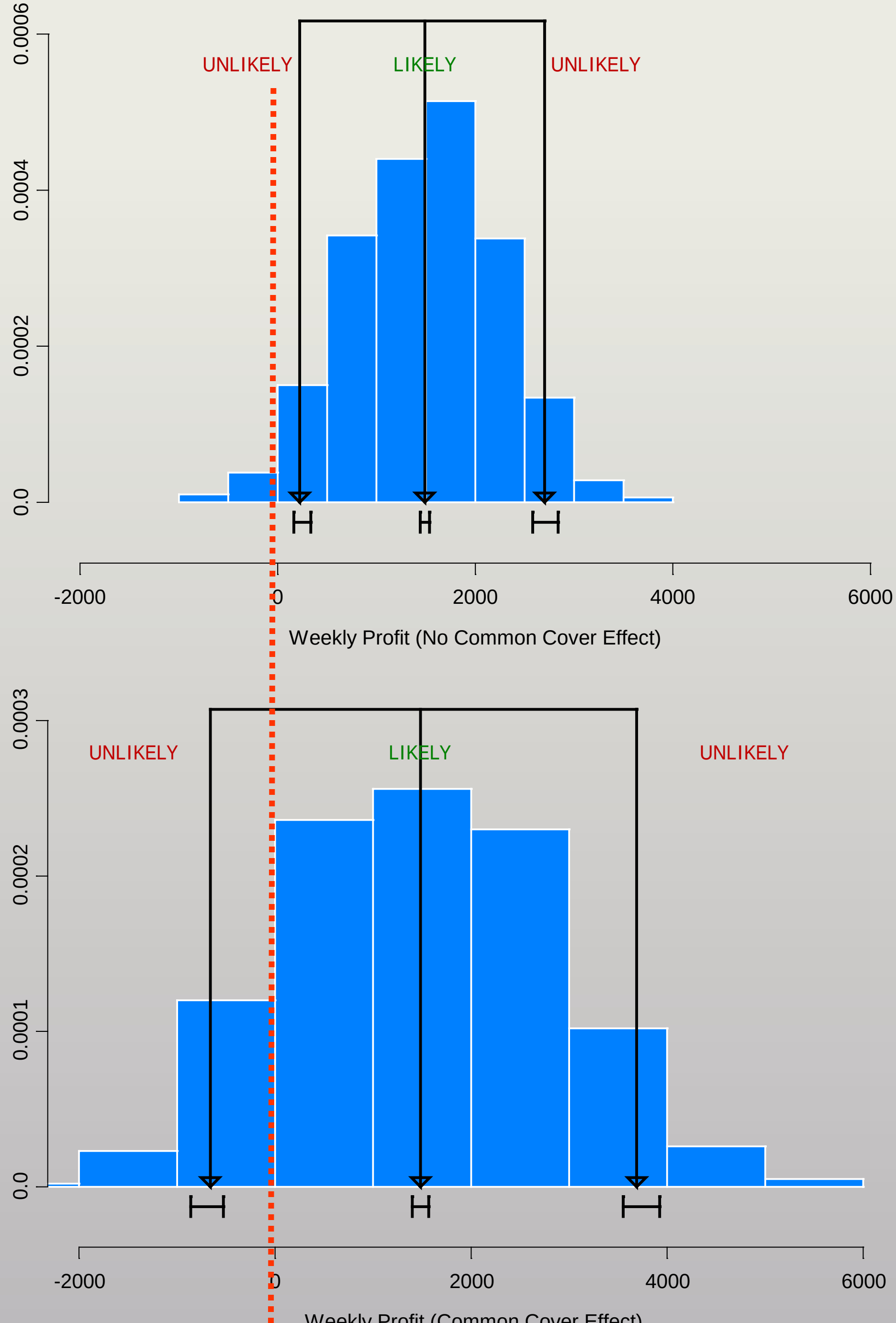






If you were going to promise an order fill time and didn't want to disappoint, what would you promise?

McCormick
Northwestern Engineering



The magazine distributor's problem

- A complicated weekly single-period inventory problem.
- Want a policy that maximizes **long-run average profit**.
- For some titles, who is on the cover matters; for others not.
- Here are simulation results for the weekly profit for the optimal policy for one of each type.

McCormick
Northwestern Engineering

Why good simulations go bad

- **We forget that we control error to measure risk**

The Ghost of R. A. Fisher (1890-1962)

- It is easy to argue that Fisher's work had profound and lasting impact on how the world does statistical experiments.
  - Factorial designs, blocking, analysis of variance, testing,….
  - All still in wide use today.
- Among other things, Fisher thought about agricultural experiments.
  - A growing season is a long time, so data are expensive.
- Classical experimental design is a structured approach to get a lot of information out of a little bit of data.

Doing simulation is not like raising corn

- Individual replications are often **cheap**.
- Response variances can **differ substantially** (even explosively) across scenarios.
- Data can be collected **sequentially** (rather than one shot) because the computer does it.
- We may want to (need to) **explore** the design space rather than identify all factors at once.
- We can (should) drive our experiment by **the error we can accept** instead of the data we can afford.
  - Asymptotic results make sense!

How I often see people doing simulation

1. Build the simulation model.
   a) Making lots of mistakes.
   b) Learning as you go along.
2. Choose an “experiment design.”
   a) Use the default number of replications in the software, or…
   b) make 30 replications.
3. Try some obvious scenarios that…
   a) Confirm what you thought, so you are done.
   b) Make you rethink what you thought would work so you try other scenarios.

There is nothing Fisher-like about this. Why shouldn’t experiment design be correct <u>and</u> support what we want to do?

McCormick
Northwestern Engineering

error level	0.1
max alternatives	26
practical difference	3.8
initial sample size	10
seed 5	
seed 2	
seed 3	

Go

total sample size	0

# machines at each workcenter	A	3	4	3	3
	B	2	2	3	2
	C	4	4	4	5
	D	3	3	3	3
	E	1	1	1	1
	replications				
	status	new	new	new	new
	threshold				
	overall mean				
	variance				
	data				

Why good simulations go bad

- We forget that we control error to measure risk

- **We use old experiment designs or no design at all**

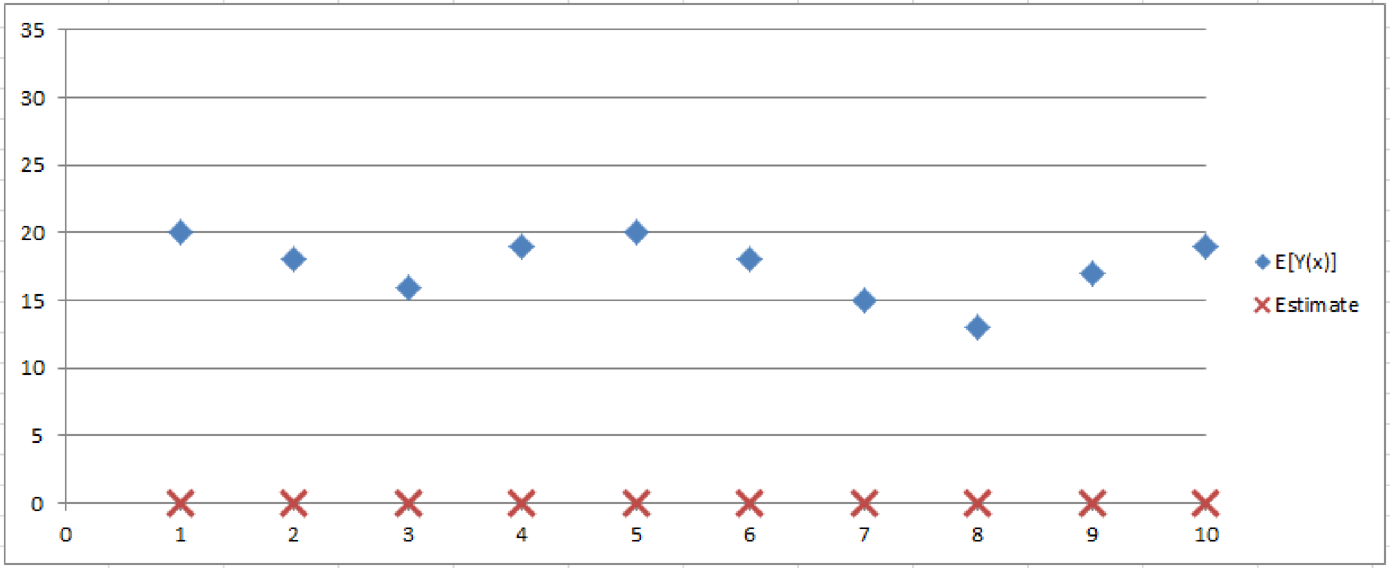
Clean up your mess: Simulation optimization

- Since mankind built its first simulation out of rocks, sticks and animal dung, there has been a primal desire to treat the simulation as the **objective function of an optimization.**
  - How many and which redundant components to include to maximize long-run system availability subject to a budget constraint.
  - Set red, green and left-turn-arrow cycles lengths to minimize mean aggregate driver delay.
  - Decide how many of each product variant to stock to maximize the expected value of profit.

Simulation optimization is hard

- Objective function is implicit in the simulation code (often no known properties).
- Objective function is evaluated with noise.
- Possible mix of integer, continuous and categorical decision variables.
- Evaluation of the objective function takes from seconds to hours.
- What can be done?
  - Metaheuristics
  - Ranking & selection
  - Adaptive random search
  - Steepest descent with stochastic gradient estimates

x	1	2	3	4	5	6	7	8	9	10
E[Y(x)]	20	18	16	19	20	18	15	13	17	19
Estimate	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Reps	0	0	0	0	0	0	0	0	0	0



Reset

R & S

Global

Local

McCormick
Northwestern Engineering

Errors in simulation optimization

1. You don’t find the optimal solution.
2. You don’t recognize the best solution you actually simulated.
3. You have a misleading idea about the value of the solution you actually did select.

It is hard to do anything general-purpose about #1, but we don’t have to accept #2 and #3.

McCormick
Northwestern Engineering

Statistical “clean-up”

- Once the optimization stops, we have a finite number of simulated solutions.
  - We solve #2 if we find the best of these.
  - We solve #3 if we control the estimation error.
  - And we have a warm start.
- Statistical “clean up” adds just enough additional simulation to guarantee, say with 95% confidence…

  - Selected solution is the best or within δ of the best of solutions we simulated.
  - Selected solution’s estimated value is correct to within ±δ.

Why good simulations go bad

- We forget that we control error to measure risk

- We use old experiment designs or no design at all

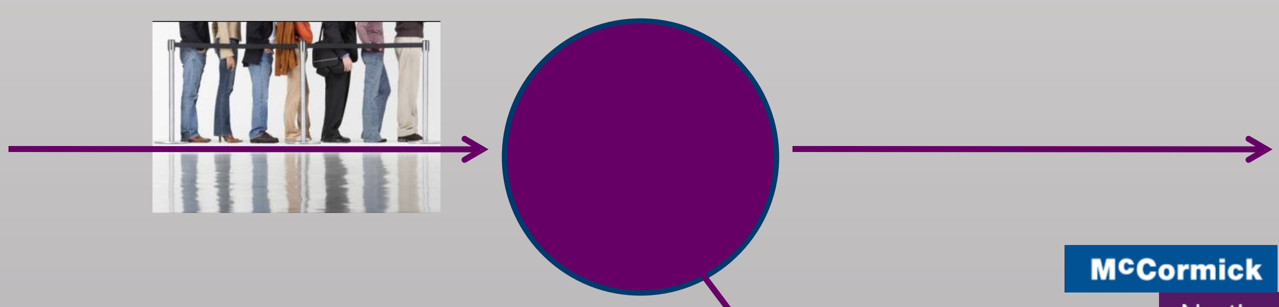
- **We settle for what the optimizer gives us**

Everything I told you is wrong: M/M/∞ Queue

Let Y be the steady-state number of customers in an $M/M/\infty$ queue with (true, real-world) arrival rate λ_c and mean service time τ_c.

Then $\mathrm{E}(Y) = \lambda_c \tau_c$

Thought Experiment: What if we did not know this result, but wanted to estimate it by observing real-world arrivals and services, then simulating the queue?



McCormick
Northwestern Engineering

Stylized simulation (sort of like we teach it)

1. Observe m real-world interarrival times $A_1, A_2 \ldots, A_m$ and estimate λ_c by $\widehat{\lambda} = 1/\bar{A}$.
   Observe m real-world service times $S_1, S_2 \ldots, S_m$ and estimate τ_c by $\widehat{\tau} = \bar{S}$.
2. Simulate n replications of the queue; on each take a single observation of the number of customers in the system in steady state: $Y_1, Y_2, \ldots, Y_n$.
3. Estimate the steady-state expected number in the queue by the sample mean, $\bar{Y}$.

Remember *m* and *n*

McCormick
Northwestern Engineering

Properties

$$\mathrm{E}[\bar{Y}] = \overbrace{\frac{m}{m-1}}^{\text{bias}} \lambda_c \tau_c$$

These results integrate over **both** simulation and input uncertainties

$$\mathrm{Var}[\bar{Y}] \approx \underbrace{\frac{\lambda_c \tau_c}{n}}_{\text{simulation variability}} + \underbrace{\frac{2(\lambda_c \tau_c)^2}{m}}_{\text{input uncertainty}}$$

Typically we focus on managing this term without thinking about these terms. Is there any way to account for this input uncertainty?

Real World with distribution $\mathcal{F}$

Real-world data $A_1, A_2, \ldots, A_m$ → Input Model F

$A_1^1, A_2^1, \ldots, A_m^1$ → F_1

$A_1^2, A_2^2, \ldots, A_m^2$ → F_2

$\vdots$

<u>Thought Experiment:</u> What would happen if…?

Empirical World with distribution $A_1, A_2, \ldots, A_m$

$A_1^{(1)}, A_2^{(1)}, \ldots, A_m^{(1)}$ →

$A_1^{(2)}, A_2^{(2)}, \ldots, A_m^{(2)}$ →

$\vdots$

<u>"Bootstrap" Experiment:</u> Simulate what happens if…?

Why good simulations go bad

- We forget that we control error to measure risk

- We use old experiment designs or no design at all

- We settle for what the optimizer gives us

- **We ignore a big risk: input uncertainty**

Efron’s simplified history/future of statistics

- **Age of Quetelet**
  - Large census-level data sets brought to bear on simple but important questions (e.g., Is the rate of insanity increasing?).
- **Classical period of Pearson, Fisher, Neyman, et al.**
  - Optimal inference for wringing every drop of information out of small scientific data sets to answer simple questions (e.g., Is Treatment A > Treatment B?).
- **Era of scientific mass production**
  - Massive data sets are generated by teams of scientists, with thousands of estimates or hypotheses that need to be answered simultaneously (e.g., microarrays)

Nelson's simplified history/future of simulation statistics

- **Automated collection of performance measures**
  - In simulation languages, if it's a resource, then calculate utilization; if its a queue, then calculate mean waiting time.
- **Simulation as an experiment**
  - Easy management of scenarios; control number of replications or run length; report confidence intervals on all results.
- **Experiment design & analysis driven by decisions**
  - Measures of error and **risk** (incl. **input uncertainty**) are standard.
  - Design attains **acceptable error** or **best use of available time**.
  - Designs support the **way people actually do simulation.**
  - Optimization with convergence guarantees and **clean up.**