Computers are Changing the (Inferential) Game: How You Can Modernize Your Mathematical Statistics Course

Peter E. Freeman - Department of Statistics & Data Science - Carnegie Mellon University

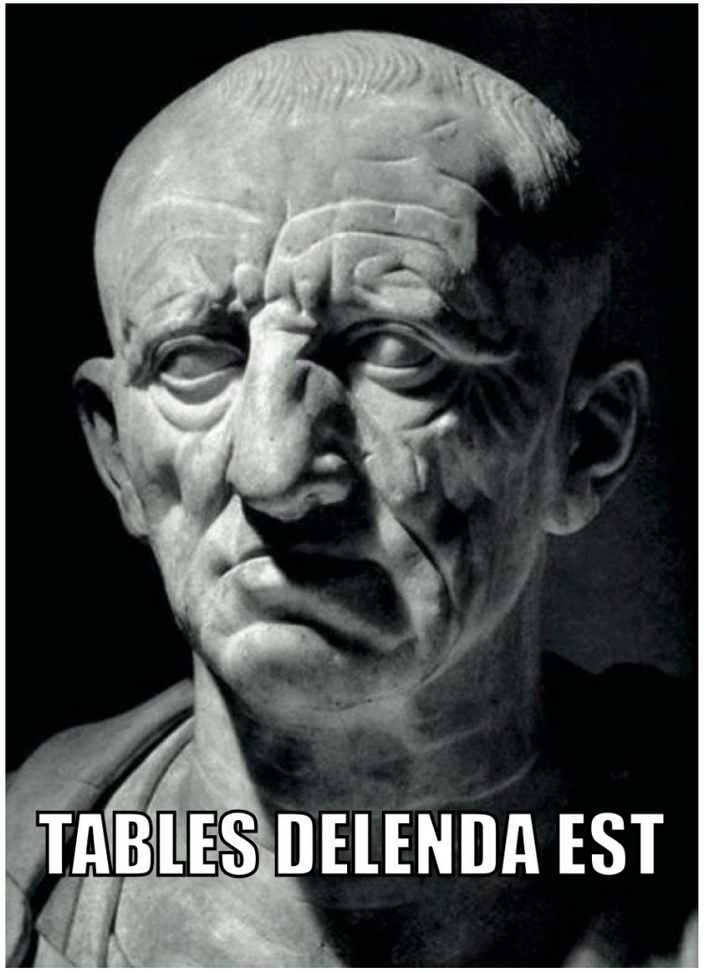
2026-06-15

Who Am I?

- I am an associate teaching professor in, and the director of the undergraduate program of, Carnegie Mellon's Department of Statistics & Data Science.
- My teaching is primarily concentrated in two areas:
  - mathematical statistics; and
  - statistical learning.
- I have been teaching mathematical statistics since 2011, and in 2022 I developed a new course sequence, *Probability and Statistical Inference I* and *II*, which utilizes a spiral-learning framework (so that I'm teaching "probability *for* statistics" as opposed to "probabilty, *then* statistics").
- The development of the material in this breakout session is motivated by my continually asking the question "why do we do things this way?"
  - A key takeaway point: always ask this same question when you teach mathematical statistics!

The Overarching Point of This Breakout Session

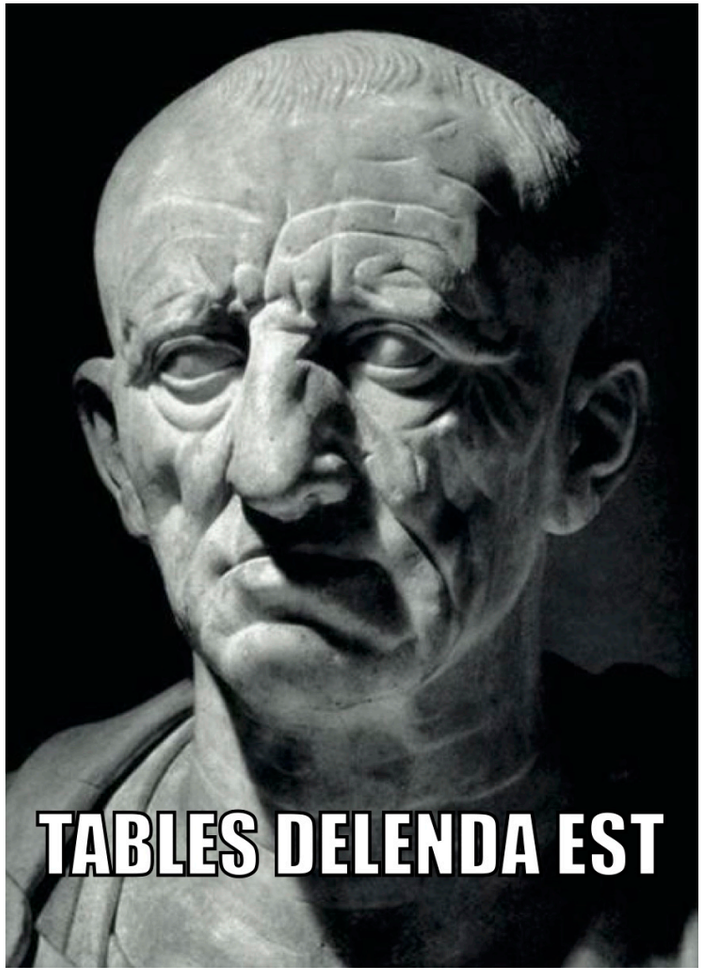
- The term “mathematical statistics” denotes the calculus-based course sequence that covers probability and statistical inference over the course of an academic year.
- Problem solving in commonly used mathematical statistics textbooks is done via analytical computations, some of which are combined with the use of statistical tables.
    - **BUT**: What should we do when students are faced with problems that they cannot solve analytically? Simply declare them “unsolvable” and walk away?
    - **AND**: Why should we teach students how to use tables when they will, in all likelihood (pun not intended), never use tables in real-life situations?



Cato the Elder hated both Carthage and statistical tables. Who knew?

The Overarching Point of This Presentation

- In this breakout session, we provide guidance on how your students can utilize numerical methods to solve inferential problems: e.g., root-finding, simulations, optimization.
    - **IMPORTANT POINT #1**: Numerical methods are meant to supplement students' analytical work and are not meant to replace entire problem-solving workflows with black boxes!
    - **IMPORTANT POINT #2**: As this is a 30-minute session, we can only cover a few examples where we utilize numerical methods. At the end, we will point you to resources that contain more information.



Cato the Elder hated both Carthage and statistical tables. Who knew?

Example 1: Transformations

- Let's assume that you are given n independent and identically distributed data $\mathbf{X} = \{X_1, \ldots, X_n\}$ that are sampled according to a normal distribution with unknown mean μ and unknown variance σ^2.
- And let's suppose that we wish to make inferences about σ^2, given the sample variance S^2.
- What statistic have you directed your students to use, and why?

  a. S^2

  b. $(n-1)S^2/\sigma^2$

- The "why" is something for you to ponder, unless you want to shout out an answer, or put an answer into the chat.

Example 1: Transformations

- The standard textbook approach is to use $(n-1)S^2/\sigma^2$, because…
  - …this quantity is distributed according to a chi-square distribution for $n-1$ degrees of freedom, and…
  - …in standard textbooks, the quantiles for the chi-square distribution are provided in statistical tables.
- What is the distribution for S^2?
  - You can have students derive this:

$$S^2 \sim \text{Gamma}\left(\frac{n-1}{2}, \frac{2\sigma^2}{n-1}\right)$$

  - (This assumes the "shape-scale parameterization.")
  - When making statistical inferences about σ^2, students would make use of, e.g., `pgamma()` and/or `qgamma()`, and *not* tables in a book.

Example 1: Transformations

- Key takeaway points:
  - Textbook approaches to problem solving are often constructed with statistical tables in mind.
  - It is important to separate the “pedagogical benefits of discussing transformations” from “how can we actually solve problems.”
    - For instance, you might have deep-seated pedagogical reasons for wanting students to work with the quantity $(n-1)S^2/\sigma^2$ in your class. And that’s perfectly fine.
    - *But when it comes time to actually solve problems*, the best approach is to have students work with S^2 directly. (For instance, try computing test power with $(n-1)S^2/\sigma^2$ as your statistic as opposed to S^2. It’s considerably more painful!)

Example 2: Interval Estimation

- Let's construct a 95% two-sided interval estimate for σ^2, given that $n = 10$ and $S^2 = 16$.
- How would you have students begin, and why?

  a. You would have them work with S^2.

  b. You would have them work with $(n-1)S^2/\sigma^2$.
- Like before, the "why" is something for you to ponder, unless you want to shout out an answer, or put an answer into the chat.

Example 2: Interval Estimation

- We've been taught to utilize the quantity $(n-1)S^2/\sigma^2$ because it is a "pivotal quantity."
  - But why do we need pivotal quantities? Because they lead us down a computational path that ends up at…(wait for it)…statistical tables.
- In the paper *Modernizing How Students Compute Interval Estimates in Mathematical Statistics Courses* (soon to be published in *JSDSE*, inshallah), we propose that teachers work with a historically underutilized approach, which is to solve the equation

$$F_Y(y_{\text{obs}}|\theta) - q = 0$$

for θ, where

  - Y is our chosen statistic (here, S^2; note that Y can be a pivotal quantity too);
  - y_{obs} is the observed value for the chosen statistic (here, 16);
  - θ is a population parameter (here, σ^2);
  - $F_Y(\cdot)$ is the cumulative distribution function for Y; and
  - q is an appropriately chosen distribution quantile (here, $1-\alpha/2$ for a lower bound, and $\alpha/2$ for an upper bound…see Section 1.16 of MPSI for more details about how q is determined).

Example 2: Interval Estimation

- Key takeaway points:
  - It is not necessary to identify a statistic that is a pivotal quantity in order to construct interval estimates!
  - The equation given on the previous slide can be solved numerically using R's `uniroot()` function…as we will show on the next slide.

Example 2: Interval Estimation

- In the code chunk below, the function `f()` represents the computation of $F_Y(y_{\text{obs}}|\theta) - q$.
  - `uniroot()` tries different values of `sigma2` within the stated `interval` until `f()` returns zero (within machine tolerance).

```
S2 <- 16
n  <- 10
f <- function(sigma2, y.obs, n, q) {
  pgamma( y.obs, shape=(n-1)/2, scale=2*sigma2/(n-1) ) - q
}
uniroot(f, interval=c(0.01,100), y.obs=S2, n=n, q=0.975)$root
```

```
[1] 7.569904
```

```
uniroot(f, interval=c(0.01,100), y.obs=S2, n=n, q=0.025)$root
```

```
[1] 53.32562
```

- Our 95% two-sided interval is $[7.570, 53.33]$. This interval is *exact* (within machine tolerance) and not an approximation!
- (Note that the computation time is negligible.)

Example 2: Interval Estimation

- Key takeaway points:
  - Given exact sampling distributions, numerical root-finders return exact interval bounds, within machine tolerance. *We are not approximating!*
  - Numerical approaches broaden the space of solvable problems *and* makes achieving solutions easier for students with less mathematical skill.
  - The use of `uniroot()` only replaces that part of interval estimation wherein we invert the sampling distribution cdf.
    - Students may still have to derive the cdf for individual data, and/or derive the sampling distribution via the method of moment-generating functions, and/or compute the expected value of the statistic (as that dictates the value of q), etc.
  - And again, while some might argue that teaching the pivotal method has pedagogical benefits, we need to separate those benefits from how we would actually go about solving problems (which here means use R and not tables).

Example 3: Point Estimation

- Assume that we are given $n = 10$ iid data sampled according to a Gamma(a, b) distribution, with probability density function

$$f_X(x|a,b) = \frac{x^{a-1}}{b^a} \frac{\exp(-x/b)}{\Gamma(a)} ,$$

  where $\Gamma(\cdot)$ is the gamma function and where both a and b are unknown.

- How would you have students determine the point estimate $\hat{a}$?

  a. Via maximum-likelihood estimation

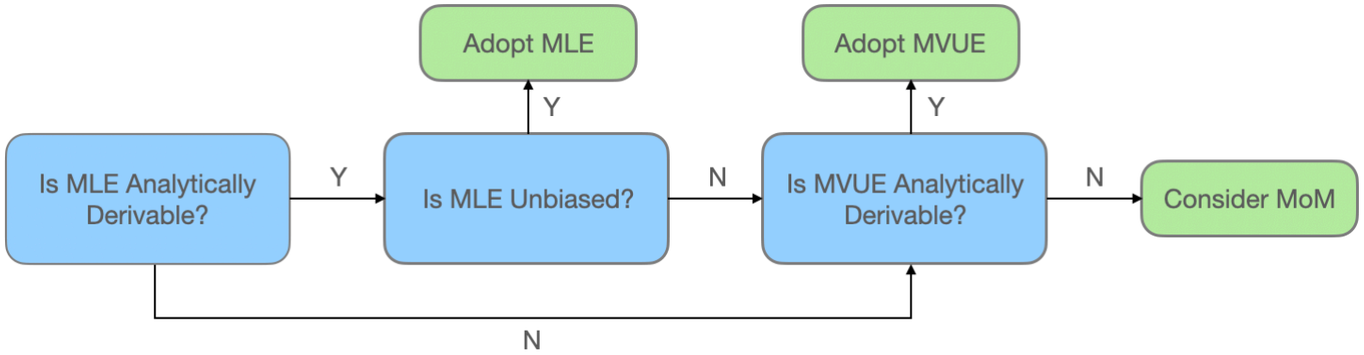
  b. Via minimum-variance unbiased estimation

  c. Via method-of-moments estimation

- And why? You know the drill…

Example 3: Point Estimation

- Here is our take on a “traditional” point estimation workflow:



- By “consider MoM,” we mean…
  - …if we have derived a biased maximum-likelihood estimator (MLE), we would compare its mean-squared error against that of the method of moments estimator (MoM), and adopt the one with a lower MSE;
  - If we cannot analytically derive the MLE or the minimum-variance unbiased estimator (MVUE), we would just adopt the MoM.
- We note that the MoM only outperforms the MLE in limited circumstances, ones that do not arise in typical undergraduate mathematical statistics class settings. We return to this point below.

Example 3: Point Estimation

- Assume that we are given $n = 10$ iid data sampled according to a Gamma(a, b) distribution, with probability density function

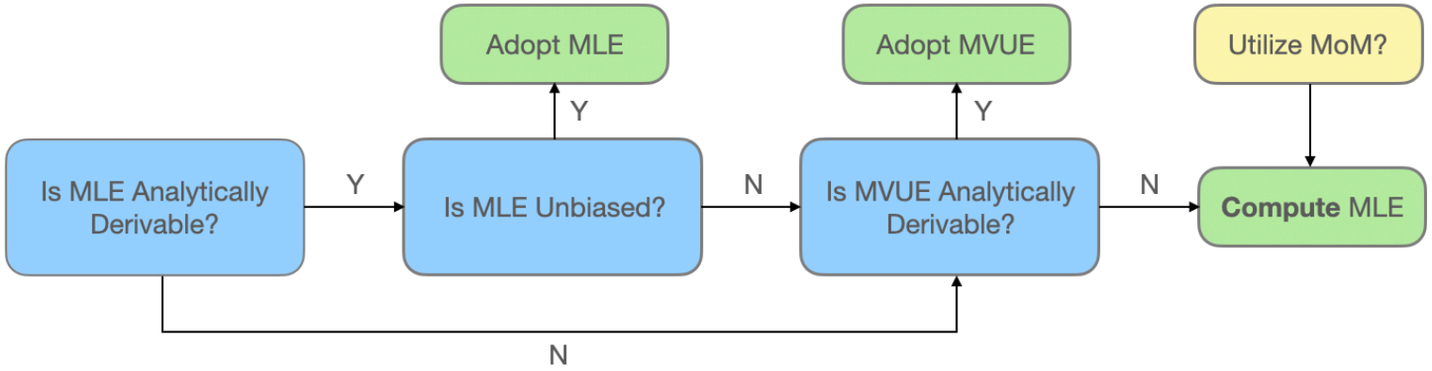
$$f_X(x|a,b) = \frac{x^{a-1}}{b^a} \frac{\exp(-x/b)}{\Gamma(a)} ,$$

  where $\Gamma(\cdot)$ is the gamma function and where both a and b are unknown.
- How would you have students determine the point estimate $\hat{a}$?

- In this example…
  - we *cannot* analytically differentiate the log-likelihood function (because we cannot analytically differentiate $\Gamma(a)$), and
  - we *cannot* analytically derive the MVUE, and so
  - we would fall back upon the MoM estimator.

Example 3: Point Estimation

- Here is our take on a better point estimation workflow:



- If we cannot analytically derive either the MLE or the MVUE, just determine the MLE via numerical optimization!
  - One can still derive the MoM, but we feel that this is now **optional** since it is rarely competitive.
  - However, we will note that MoM estimates are good initial parameter values to feed to a numerical optimizer.

Example 3: Point Estimation

- Let's assume we know neither a nor b.

```
g <- function(x, par) {
  -sum( log( dgamma(x, shape=par[1], scale=par[2]) ) )
}
par <- c(1, 1)
round(optim(par, g, x=X)$par, 3) # X is the data vector
```

```
[1] 2.516 3.967
```

- We find that $\hat{a} = 2.516$ and $\hat{b} = 3.967$.

Example 3: Point Estimation

- Key takeaway points:
  - Textbook approaches to problem solving are constructed without numerical approaches in mind.
    - One consequence of this is that we conventionally teach method-of-moments estimation when we do not actually need to do so.
  - Numerical approaches broaden the space of solvable problems *and* makes achieving solutions easier for students with less mathematical skill.
  - Numerical optimization simply replaces differentiation of the log-likelihood function: by using it, we are not “replacing” the full maximum likelihood estimation workflow with a “black-box.”
    - In other words, students still have to learn what the MLE *represents*, and what its properties *are*, etc.
    - Students can expand on the code given on the previous slide to determine estimator bias and variance, etc., via simulations, and possibly compare these quantities against those found for the MoM estimator.

Conclusions and Additional Resources

- At the risk of sounding like a broken record:
  - Textbook approaches to problem solving are often constructed with statistical tables in mind.
  - Textbook approaches to problem solving are constructed without numerical approaches in mind.
  - Numerical approaches broaden the space of solvable problems *and* makes achieving solutions easier for students with less mathematical skill.
- Ultimately: question *everything*. And have your students use R.

- Some additional (biased, but still) resources you mind find useful:
  - The freely available textbook *Modern Probability and Statistical Inference: Illustrated with R*, at https://mpsibook.github.io
  - Supporting material for the 2024 ECOTS presentation *Modernizing the Teaching of Confidence Intervals in Mathematical Statistics Courses*, at https://github.com/pefreeman/Confidence-Intervals
  - Supporting material for the 2025 USCOTS poster *Multinomial Simulations: Why We Can (and Should) Use Them Instead of the Chi-Square Goodness-of-Fit Test*, at https://github.com/pefreeman/USCOTS-2025
  - Supporting material for the 2026 ICOTS presentation *The Future of Statistical Inference in the Classroom: Aligning Theory with Computation*, at https://github.com/pefreeman/ICOTS-2026
- The slides for this session are available at https://github.com/pefreeman/ECOTS-2026
- Ideas? Questions? `pfreeman@cmu.edu`

Extra Example: Numerical Simulation

```r
set.seed(36235)
S2 <- 16
n  <- 10
f <- function(sigma2, y.obs, n, q) {
  num.sim <- 10000
  X       <- matrix(rnorm(n*num.sim, mean=0, sd=sqrt(sigma2)), nrow=num.sim)
  Y       <- apply(X, 1, var)
  sum(Y <= y.obs)/num.sim - q
}
uniroot(f, interval=c(0.01,100), y.obs=S2, n=n, q=0.975)$root
```

```
[1] 7.631624
```

```r
uniroot(f, interval=c(0.01,100), y.obs=S2, n=n, q=0.025)$root
```

```
[1] 51.59906
```

- The code given above is identical to that given in Example 2, except here we replace `pgamma(...) - q` with the difference between the empirical cdf (`sum(Y <= y.obs)/num.sim`) and `q`.
  - The empirical cdf for S^2 itself is built up by simulating 10,000 datasets given a particular value for σ^2, and computing S^2 for each.
- Given the exact distribution for each datum, the numerical root-finder will return the exact interval bounds, within machine tolerance and sampling error. We are not approximating!
  - To reduce the sampling error, increase the number of simulations (`num.sim`). The displayed code runs in ≈ 1 CPU second. We would advise increasing `num.sim` by a factor of 10-100 in actual analyses.