Statistical Computing: Non-Ignorable Missingness in the Graduate-Level Social Science Curriculum

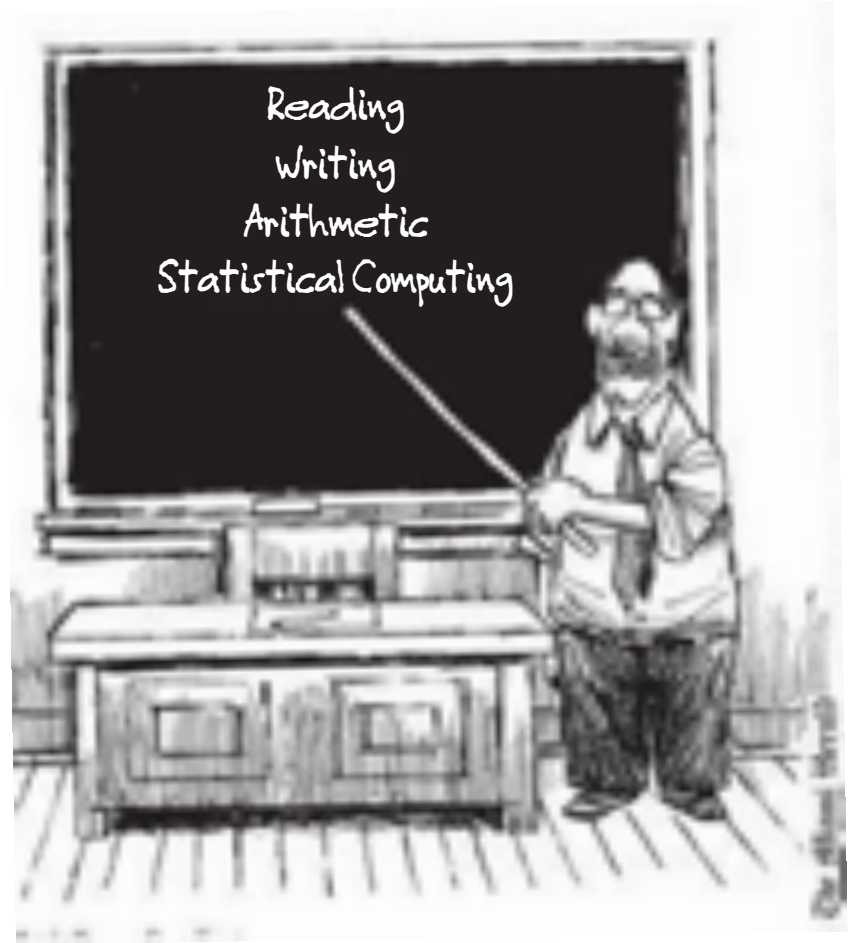
Andrew Zieffler

Department of Educational Psychology, University of Minnesota
zief0002@umn.edu

Statistical computing is an important skill for students.

"Computational literacy and programming are as fundamental to statistical practice and research as mathematics" (Nolan & Temple Lang, 2010, p. 97).

"Computing is an essential component of statistical practice and research" (Gentleman, 2004, p. 209)



What does the statistical computing curriculum currently look like in statistics departments (c. 2010)?

"Traditionally, the computing taught in the statistics curriculum falls into one of two camps, (a) numerical analysis and algorithms for statistical methods; and (b) the nuts and bolts of how to use a programming language" (Nolan & Temple Lang, 2010, p. 99).

What does statistical computing in the social sciences look like?

"Computational social science has three important features. Firstly, it often involves **big data**; data sets so large that conventional database and analysis techniques cannot handle them with ease. Secondly, dealing with these big data sets has given rise to **analysis techniques that are specially developed for big data**. Given the size of the data, resampling and cross-validation approaches become feasible that allow both data-driven exploration and checks on overfitting the data. A third important feature is **simulation**, especially agent-based simulation" (Hox, 2017, p. 3).

Survey of Quantitative Methods/Psychology Ph.D. Programs

- Examined the Ph.D. requirements for 36 programs across the United States

  - *Advanced Statistical Computing and Data Analysis* (University of Minnesota)
  - *Computer Applications to Research Methodology* (University of Pittsburgh)
  - *Computer Assisted Data Management and Analysis* (James Madison University)
  - *Computer Packages for Statistical Analysis* (University of Iowa)
  - *Computational Modeling* (Vanderbilt)
  - *Computational Statistics* (Teachers College)
  - *Computational Statistics* (University of North Carolina)
  - *Data Analysis and Simulation in R* (University of Texas)
  - *Data Analysis using Computer Packages* (University of Pittsburgh)
  - *Mathematical Foundations and Simulation Techniques* (University of Maryland)
  - *Programming for Psychometric and Statistical Modeling* (University of Massachusetts)
  - *Programming Methods in Cognitive Science* (Indiana University)
  - *R for Educational and Social Sciences* (University of North Carolina, Greensboro)
  - *Scientific Computing for Psychological and Brain Science* (Vanderbilt)

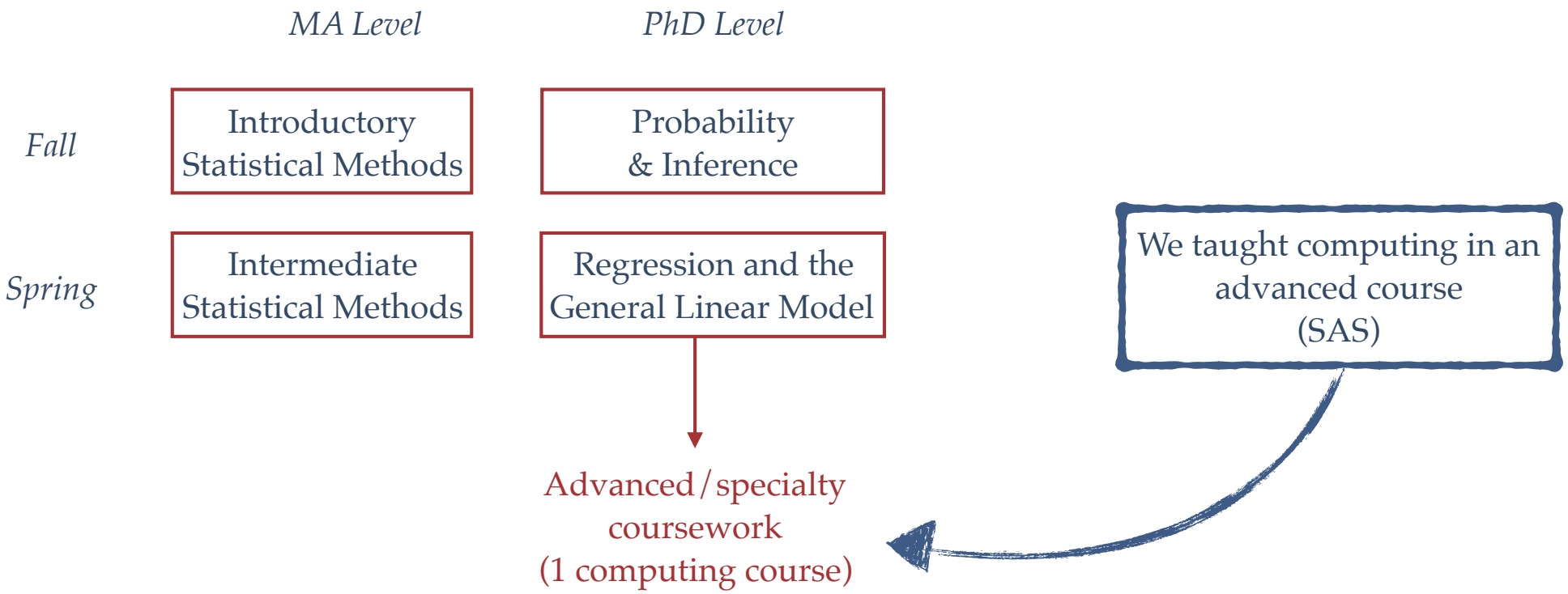
- Only 8 programs had a required computing course

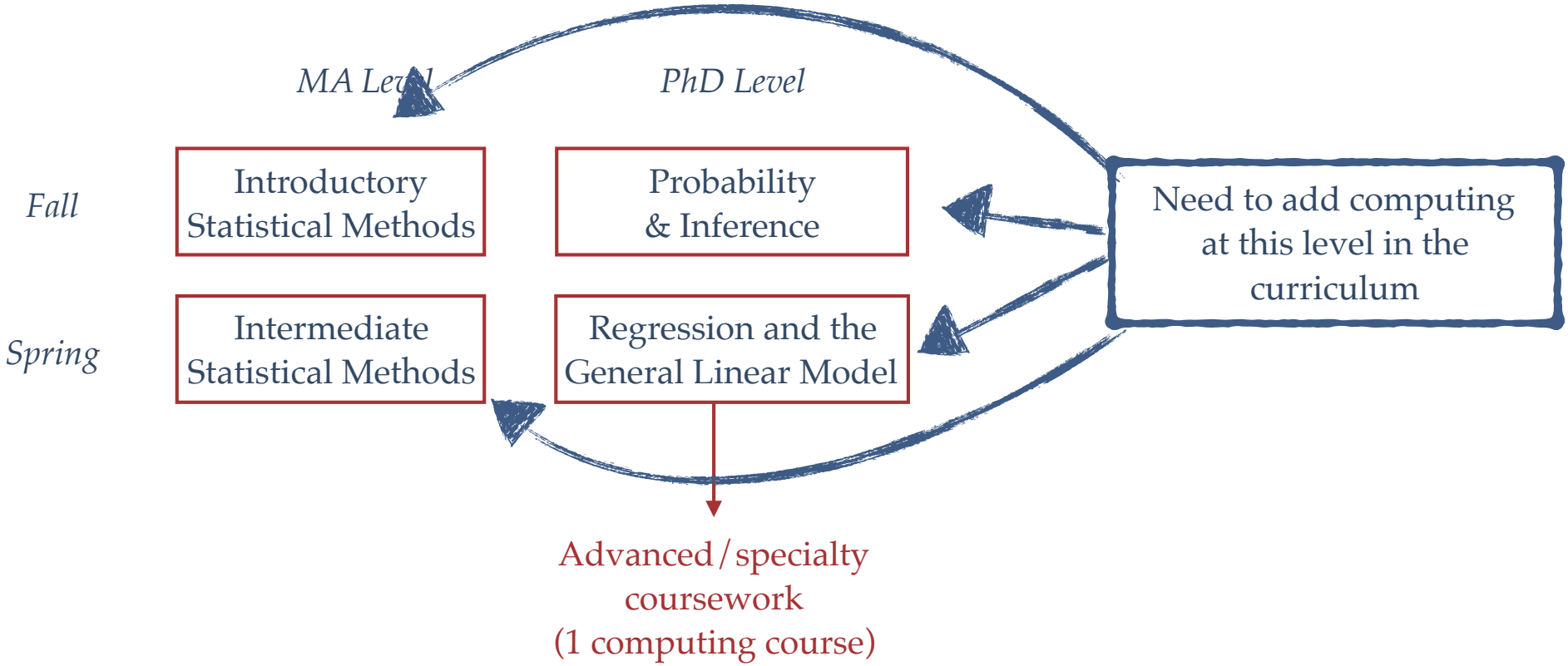
My experience with adding computing

Instructional Questions

Where and how does computing fit into the curriculum?

Quantitative Methods coursework at University of Minnesota (c. 2006)




MA Level
PhD Level
Fall
Introductory Statistical Methods
Probability & Inference
Spring
Intermediate Statistical Methods
Regression and the General Linear Model
Need to add computing at this level in the curriculum
Advanced/specialty coursework (1 computing course)

In 2008, I decided to switch software from SPSS to R
(no RStudio at this time)

Why?

1. Because R could fit ALL the models and perform analyses that we were teaching.
2. More emphasis on computing
3. Emphasize "statistical thinking" (you have to ask for what you want; no "shotgun" statistics or plots)
4. Potential to do randomization/bootstrapping
5. Could look at syntax of functions to determine what was actually being computed
6. Free
7. Because I was using R in my work

YES

I had **one other faculty member** on board at that time. He was teaching longitudinal data analysis and didn't want to pay for a SAS license any longer.

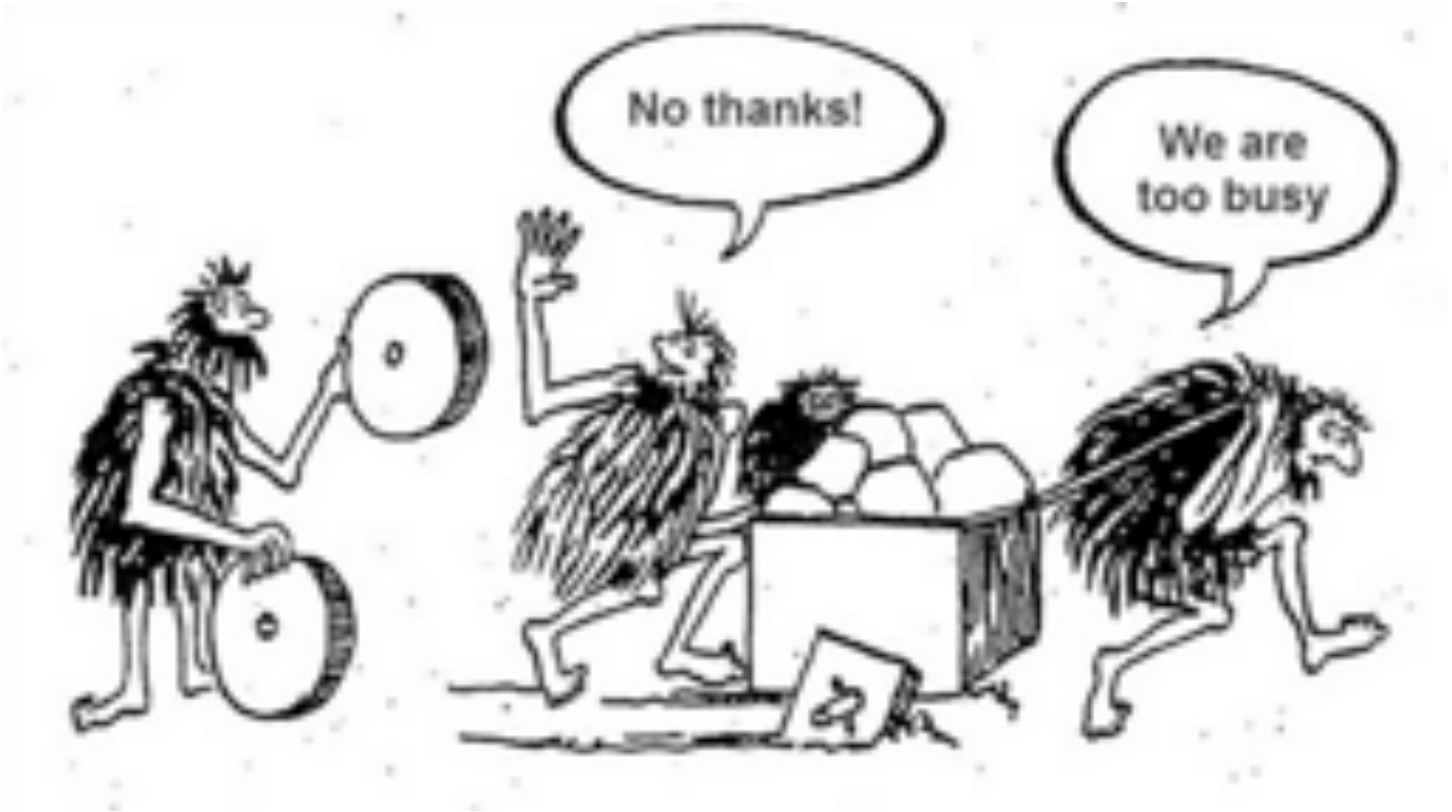
NO

Other faculty in the program

MAYBE

How do you convince others that this is of educational value?

How do you get them to take the time to learn something new well enough to teach it?



Change faculty by impacting their students.

- Students created super-nice plots, reports
- Students could do things computing-wise their advisors could not (without fancy software)
- Students began to teach their advisors
- Started a student computing club (student-driven)

Change faculty by impacting their research

- Gave free faculty/student workshops
- Held open computing office hours
- Signed up close colleagues to help teach some R workshops for the broader Twin Cities area

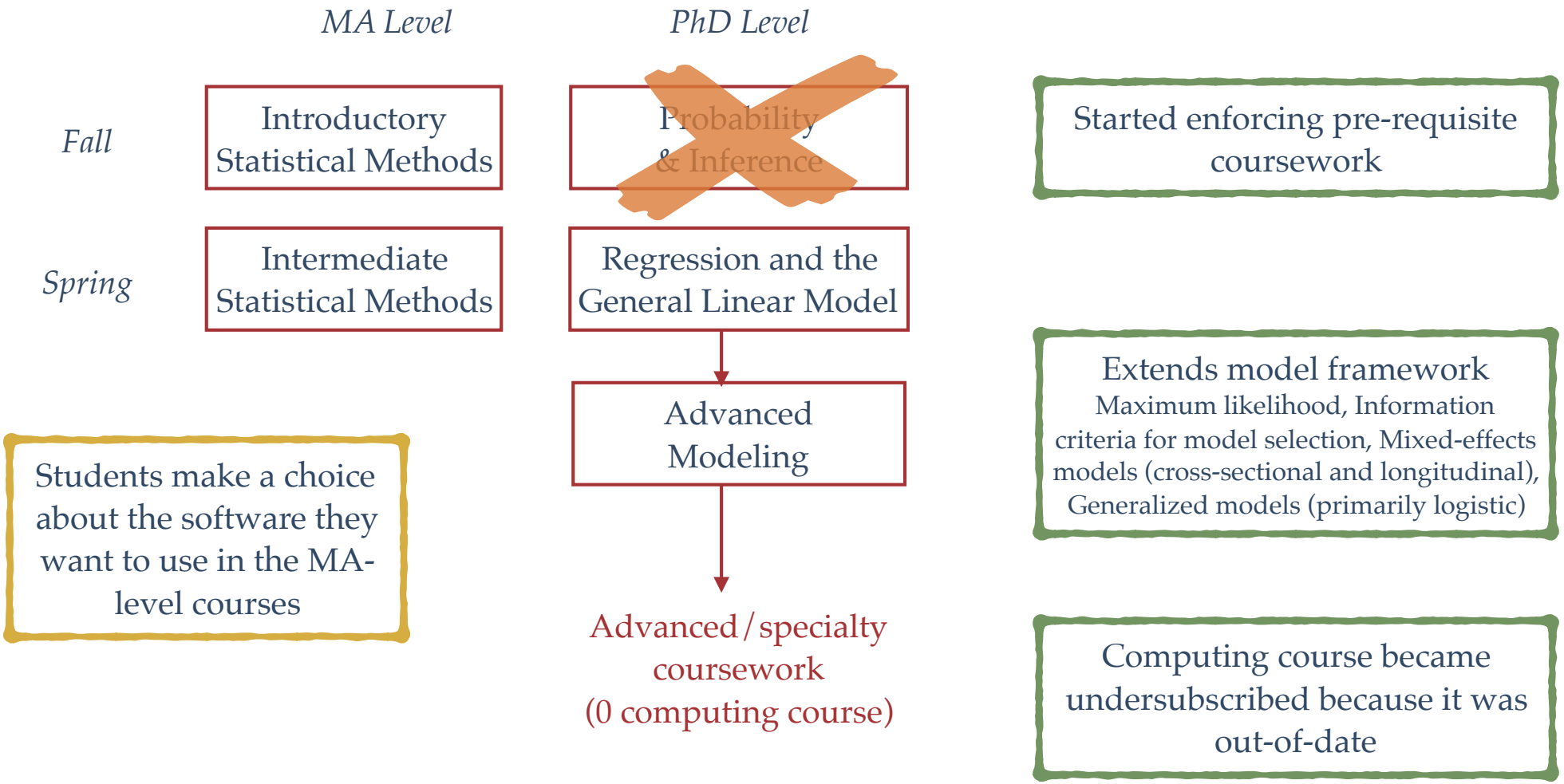
In 2011, RStudio was released and changed everything.

It has directly influenced the content of our methods courses and the success of integrating computing in these courses.

RStudio is like having a set of training wheels that helps you learn to ride a bike, but then as you get better you realize those training wheels can be used to modify your bike and help you ride better. But it doesn't stop there. Once you have become a *really* good rider, you realize that those training wheels can be turned into a rocket engine… the possibilities.

Where we are at now

Quantitative Methods coursework at University of Minnesota (c. 2018)



Instructional Questions

To what extent will we explicitly teach computing in the course vs. having students learn it on their own?

We decided to explicitly teach computing in the classroom, trying to integrate it with the statistical content.

DANGER

Students will believe that you are teaching an R/Python class; not a statistics course.

This has diminished over the years, as the computing content becomes more streamlined and consistent.

Instructional Questions

What are the computing skills that our social science graduate students need?

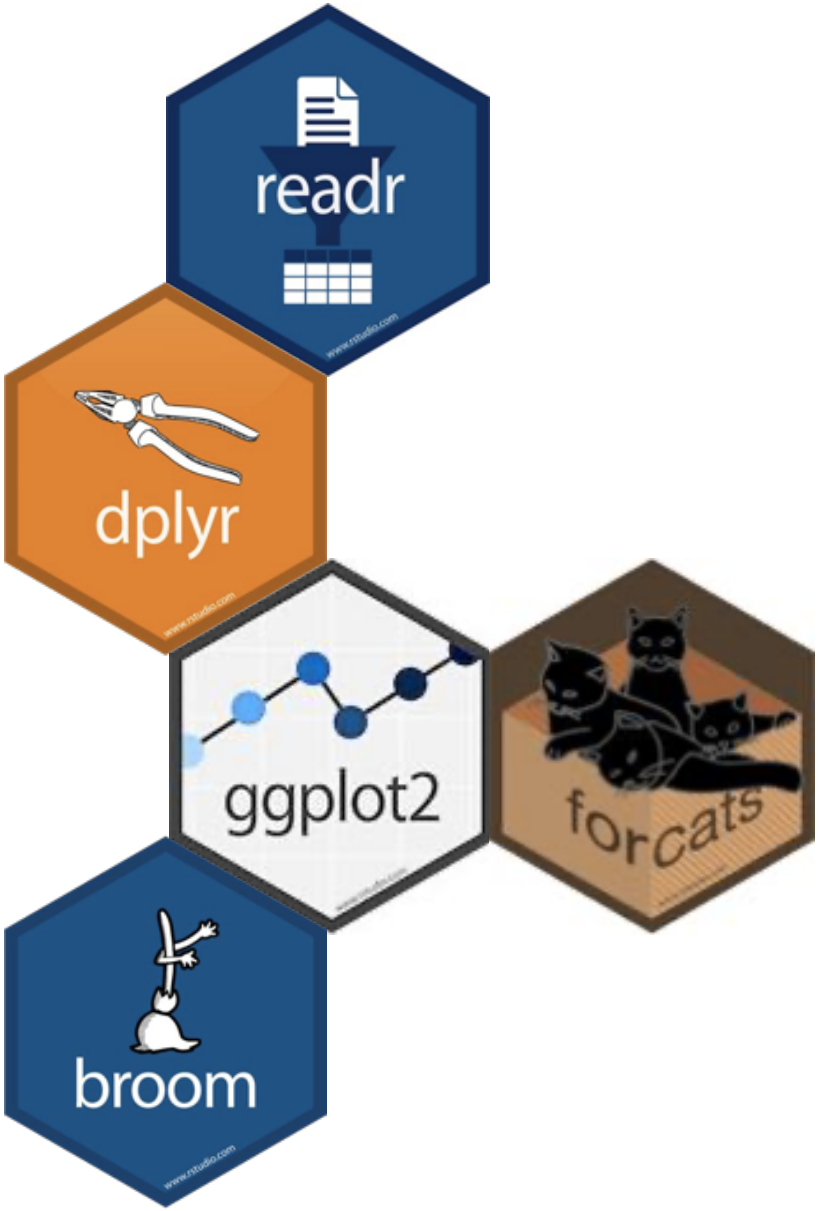
What is the right amount of computing to include in these courses?

Suggested Taxonomy of Computing Topics

Nolan & Temple Lang, 2010

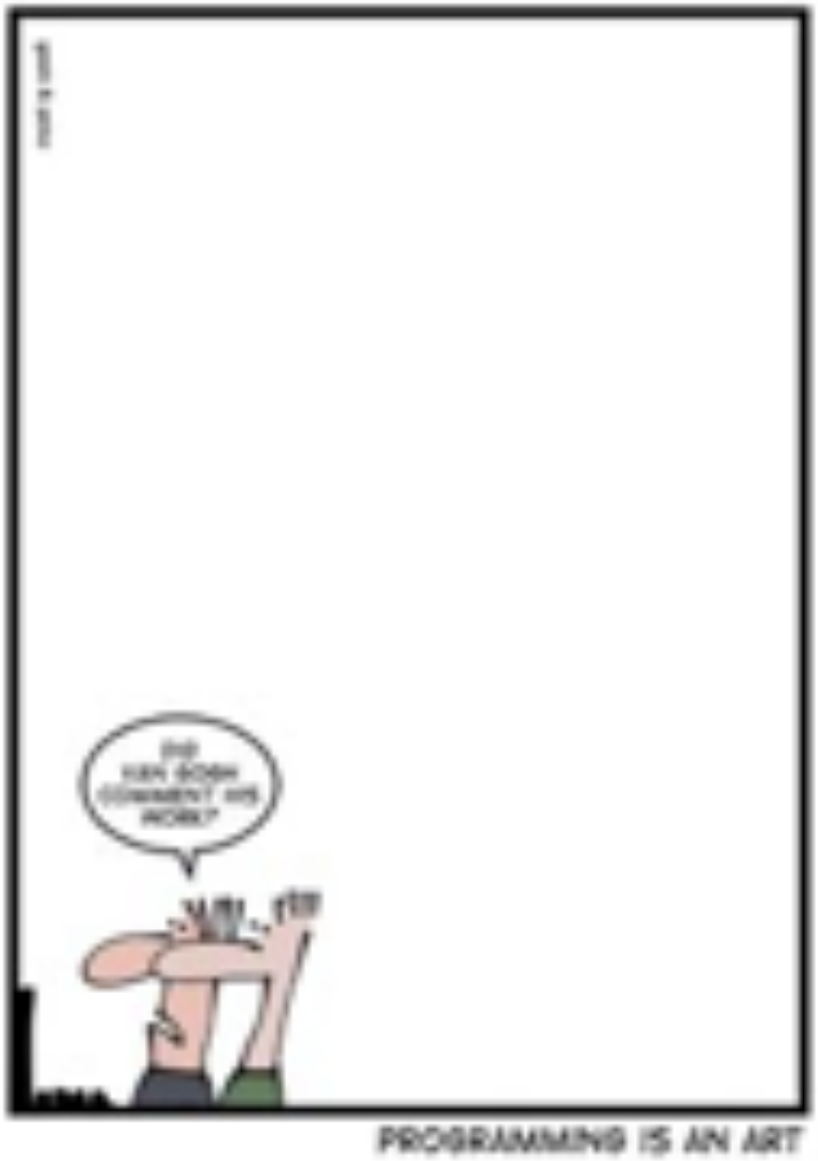
https://www.stat.berkeley.edu/users/statcur/Taxonomy/ComputingTopics.png

In our **first course** we emphasize using computing to ANALYZE data and ORGANIZE workflow (e.g., scripts).



In our **second course** we continue to build students' skills and knowledge around what they learned in the previous course, and we emphasize using computing for REPRODUCIBILITY.

The most optimum choice from a programming point-of-view may not be the most optimum choice from a pedagogical point-of-view.

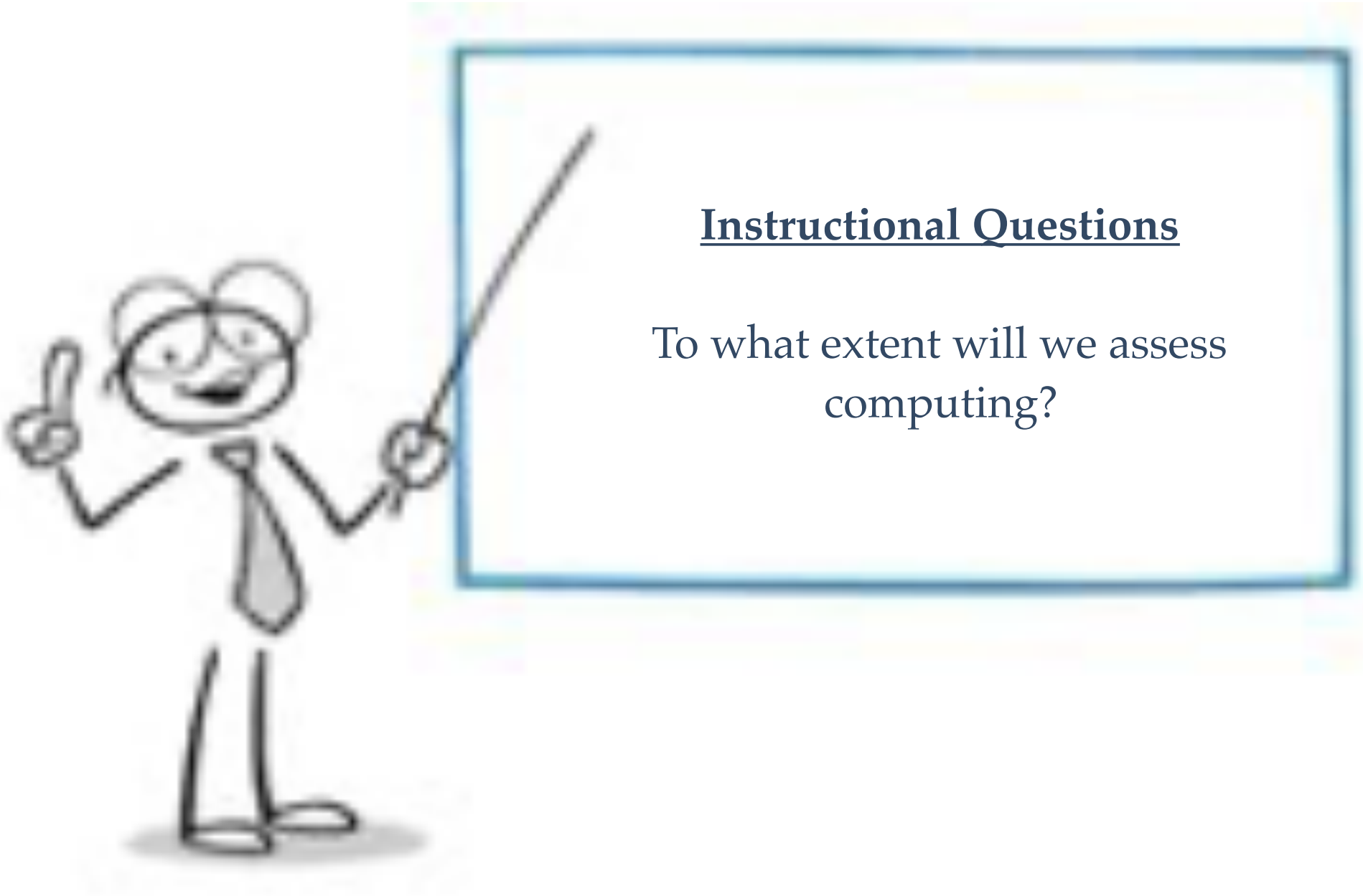


WARNING

Students will mimic whatever you model in the course. Unlearning is difficult, so your pedagogical choices matter.

If you use a particular library or set of code to do something, students will also likely use the same syntax in their work.

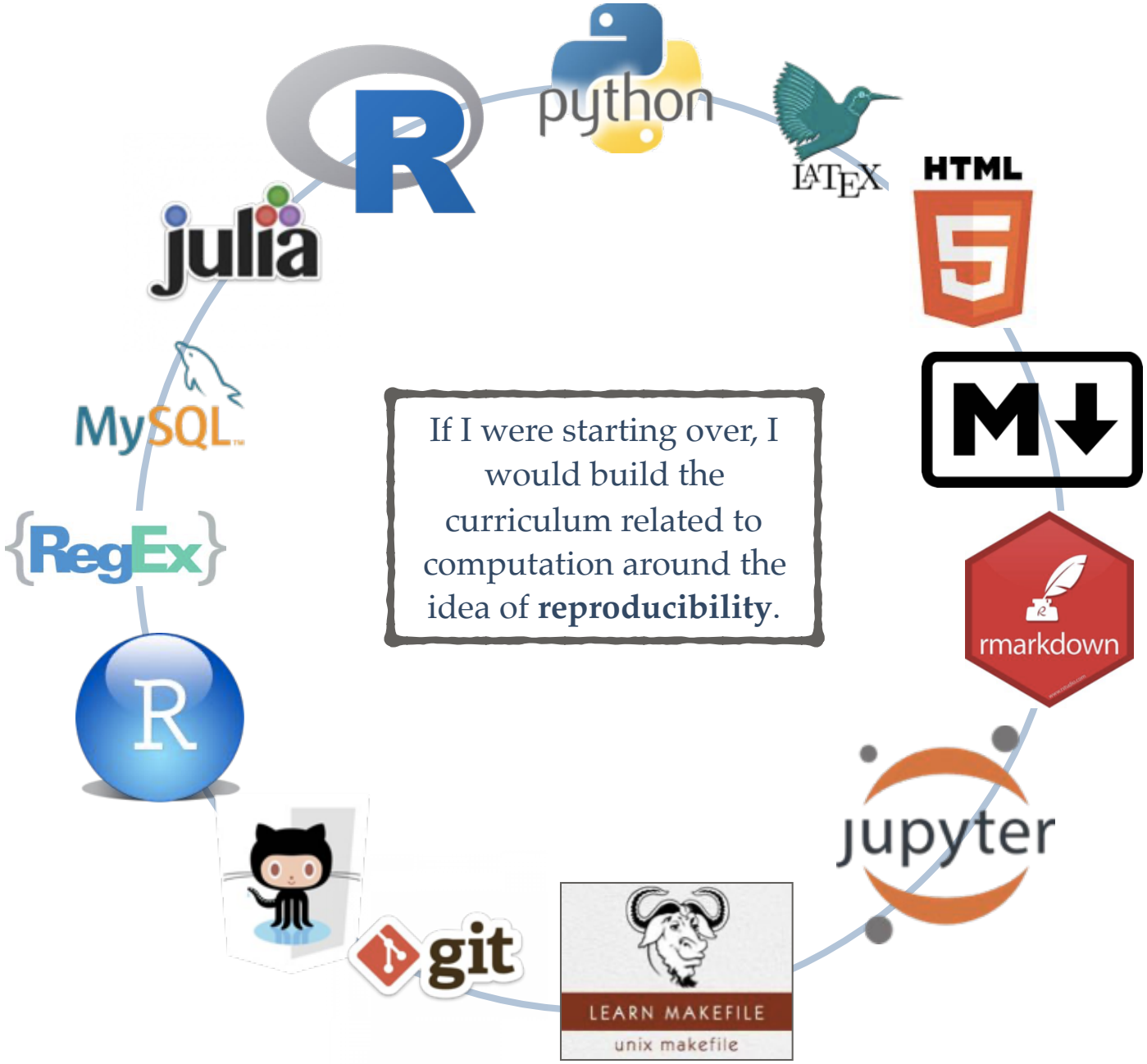
If you want them to comment their syntax, you need to comment your syntax…all the time.


Instructional Questions
To what extent will we assess computing?

Students' judgments about what content is important are directly connected to what content is assessed.

- No tests/quizzes…assignments only
- Assignments all require data analysis via computing
- Some practice, but often pushed learning beyond what was taught in the classroom
- Collaboration is encouraged

Advice for others


python
LATEX
HTML
julia
MySQL
RegEx
If I were starting over, I would build the curriculum related to computation around the idea of **reproducibility**.
rmarkdown
R
jupyter
git
LEARN MAKEFILE
unix makefile

What are the resources students have for helping them learn computation skills?

How do we "sell" this emphasis to constituent departments, content advisors, etc.?

How does this emphasis on computation affect more advanced coursework?

Advanced Placement Obedience Training

Merci!
THANK YOU!

zief0002@umn.edu

Slides available at:
http://www.datadreaming.org/

References

Gentleman, R. (2004). Perspectives on statistical computing. *The Canadian Journal of Statistics, 32*(3), 209–226.

Hox, J. J. (2017). Computational social science methodology, anyone? *Methodology, 13*(Supplement), 3–12.

Nolan, D., & Temple Lang, D. (2010). Computing in the statistics curricula. *The American Statistician, 64*(2), 97–107. https://doi.org/10.1198/tast.2010.09132