Teaching Econometrics Dynamically with R-Shiny[1]

Shawna K. Metzger, *Michigan State University, USA*

Teaching is a constrained optimization problem. There are numerous topics that we could cover in a given class, and for each of those topics, there are varying levels of depth at which we could discuss them. However, we have a finite amount of contact time with the students to do so. How should we allocate that time to give students the best chance of learning the content?

Methods courses feel the time pressure in a particularly acute way. For substantive courses in political science, preexisting knowledge about history or civics is useful but not necessary. Most of the knowledge we construct for our students in these courses comprises new information that requires minimal background knowledge for comprehension on the students' part. Contrast this with methods courses, in which the knowledge we construct for our students requires a basic grasp of algebra, at the least, if not also a minimal understanding of research design from coursework in the natural sciences and a vague recollection of probability and statistics. With time being at such a premium, how might we raise our teaching effectiveness such that students internalize more information per minute of instruction?

This article discusses how Shiny apps can help us teach methods more effectively—particularly when discussing regression models—by providing students with the ability to interact dynamically with the material. Shiny is an `R` package written by Joe Cheng of RStudio that provides functionality for instructors to produce interactive web apps using `R` known as "Shiny apps." In the ideal, the apps' interactivity allows us to teach the same content in less time than usual. This is a powerful potential efficiency gain, especially when methods courses typically are expected to cover much ground in a limited amount of time. The apps show one way to have "fundamental concepts [be] accessible" to students while "minimiz[ing] prerequisites to research," including the need to write code—a practice some statisticians argue should be adopted by undergraduate statistics courses (Cobb 2015); some social scientists argue similarly for our methods courses (Bailey 2019).

PEDAGOGICAL MOTIVATION

To get a sense of why Shiny is beneficial for teaching methods, it helps to understand what Shiny can do and why its dynamicism is important.

Why Is Shiny Helpful for Teaching Econometrics?

Simply stated, for an overwhelming number of tasks (if not nearly all), Shiny can do anything `R` can. It also can do more than `R`, which is where its power lies. Shiny can dynamically display text, Console output from `R` commands, LaTeX equations, interactive and sortable data tables, images, videos, static and interactive graphs, network diagrams, and interactive maps. Additionally, Shiny apps can accept user input through sliders, radio buttons, checkboxes, dropdowns, text input, toggle buttons, and file uploads, including previously saved `R` data files. Shiny is not the only way to code apps with this functionality. However, its attractiveness stems from it being `R`-based—a language with which many instructors are already familiar. Users require no knowledge about `R` to use the apps once they exist.[2]

Shiny's uses are myriad but some well-known examples already exist within the social sciences. Muth, Oravecz, and Gabry's shinystan package[3] is a prominent example. It uses Shiny to generate visual post-estimation diagnostics for Markov chain Monte Carlos. In `R`, users estimate a Bayesian model of interest using stan or rstanarm, as usual, but then pass the resulting model object to the Shiny app. As of this writing, shinystan's first major page contains multiple diagnostic graphs across five different tabs. Each tab's graphs are interactive, allowing the user to zoom in on specific portions, which also will auto-zoom every other graph in the tab. The app's other major pages pertain to different aspects of the model's estimates. Other Shiny app examples in political science include power calculations for survey experiments,[4] simulating the effect of electoral rules on representation,[5] and estimating the time commitment associated with a course's assigned workload.[6]

For creating methods-centric apps for pedagogical purposes, Shiny's key feature is its dynamicism: the ability to take various interactive inputs from the user, perform any number of tasks, and then output information involving the inputs. These outputs are dynamic—that is, their contents will update in real time if users make changes to the app's inputs. This dynamic, interactive aspect distinguishes Shiny from other ways of doing things in `R`. The same dynamicism provides students with an easy way of exploring content that is impossible to do with regular `R` code.

Dynamicism: So What?

Shiny's dynamicism is important because of what we know about effective teaching, particularly in a STEM context.

doi:10.1017/S1049096521001141

Although this research suggests that no "one-size-fits-all" approach to teaching exists, we do have information on general patterns that promote effective learning environments (Groth 2013, especially chs. 1–2).[7] One such pattern relates to active learning, "generally defined as any instructional method that engages students in the learning process" (Prince 2004, 223). Students engaged in active learning tend to retain more information (Ruhl, Hughes, and Schloss 1987) and to perform better on assessments of knowledge (Deslauriers et al. 2019; Freeman et al. 2014).[8] Structuring (some) tasks and activities so that students work through the problems themselves in small methodical steps—rather than being led, told, or expected to make major conceptual leaps on their own[9]—is an example of an active-learning strategy.

Shiny's key feature is its dynamicism: the ability to take various interactive inputs from the user, perform any number of tasks, and then output information involving the inputs. These outputs are dynamic—their contents will update in real time.

Active learning often appears in tandem with the idea of deep learning—which emphasizes depth rather than breadth of knowledge—with instructors promoting connectivity among concepts and sophisticated engagement rather than covering a wide array of seemingly atomized facts at a basic level (Bean 2011, ch. 9; Biggs and Tang 2011, 27). We know that students retain information longer after a course ends when they learn it at a deeper level (Bacon and Stewart 2006). We also know that prompting students to reflect on their knowledge helps them to consolidate it (Yancey 2009). Reflection can range from an informal sentence or two describing a pattern or explaining a connection to a more formal assignment. Writing can facilitate this process (Menary 2007), even in quantitative contexts (Bahls 2012). Part of consolidating knowledge involves encouraging students to make connections across topics not only within but also across courses (i.e., "integrative learning") (Huber, Hutchings, and Gale 2005; Yancey 2009). This increases the chance that they will be able to correctly recall the information later—even in new contexts.

Integrating the Apps

Shiny apps can promote active, deep, reflective learning because of the opportunity they afford students to experiment concretely with new material, subtly guided by an app's layout and any accompanying activities an instructor assigns.

How might we integrate the apps into our courses? My own usage varies but generally falls into one of the two following categories:

1. A *pre-class activity* with two or three informal questions for students to think through as they interact with an app. Otherwise, I provide little contextual information. The idea is to give students low-stakes exposure to the upcoming lecture's key ideas in a situation that fosters developing their intuition about these ideas before formalizing them. I then begin the lecture by asking students what the app seems to do and let the discussion grow organically from there.
2. An *in-class activity* in which I make brief introductory remarks before dividing students into pairs or triplets and explain the activity, typically via a worksheet. The activity usually lasts 10 to 20 minutes, depending on the quality of the side conversations I overhear as I circulate around the room. After the activity concludes and we reassemble as a class, the lecture continues as normal, with one difference: when we hit key ideas illustrated by the app, I say as much outright before asking students to describe the patterns they observed from the activity's relevant segment.

Regardless of whether the activity is pre-class or in-class, there is an important commonality. Students need to see and learn how the various pieces of information are connected in the context of a specific topic or concept before they can begin consolidating their knowledge. Part of our role as instructors is to facilitate these connections by pointing them out explicitly.

EXAMPLE: MONTE CARLO SIMULATIONS

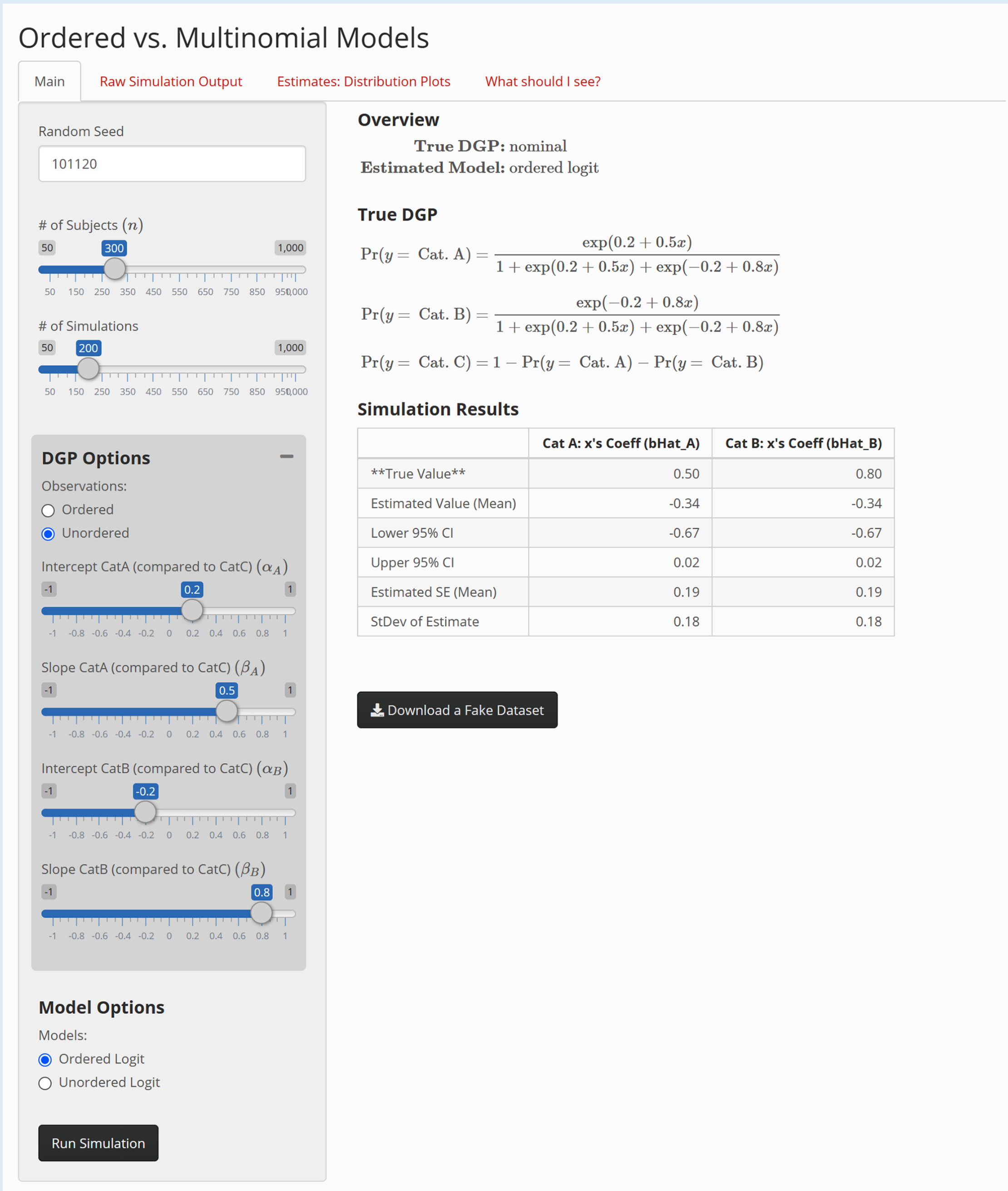
Using Monte Carlo simulations to teach regression is not new. Their pedagogical power derives from showing how an estimator performs under conditions that we, as the users, control—akin to conducting an experiment. Carsey and Harden (2013) dedicate their entire book to demonstrating how R simulations can help convey statistical concepts, ranging from regression (on this point, see also Bekkerman 2015) to model-like tests of substantive theories. The pedagogical effectiveness of simulations follows the same findings that we have from research on the effectiveness of other class activities: they help students grasp the content better, provided that the activities are well thought out, have clear goals (and that students are aware of these goals), have clear ties to broader course learning outcomes (and that students

Shiny apps can promote active, deep, reflective learning because of the opportunity they afford students to experiment concretely with new material, subtly guided by an app's layout and any accompanying activities assigned.

https://doi.org/10.1017/S1049096521001141 Published online by Cambridge University Press

Figure 1

ord_mnl, "Main" Tab



Note: Package version: Shiny_1.5.0.

are aware of these ties), and students are prompted to reflect on the activity once it is completed.[10]

Using Shiny to facilitate Monte Carlo simulations is new.[11] Using an app to set up the simulations removes the need for students to be aware of the simulation code—or to program in R at all, for non-R courses. Instead, students can focus on the pedagogically salient points. In my Cambridge Element, I discuss two such simulation apps involving ordinary least squares (Metzger 2021).[12]

However, simulation-based apps about other models are possible. I focus on one of Carsey and Harden's (2013, sec. 6.4.1) examples suitable for a maximum likelihood estimation/generalized linear models course: the difference between ordinal and multinomial data and the ramifications of these differences for regression. Their code serves as the basis for the ord_mnl app (figure 1). I usually assign this app for the first multinomial-model lecture, which occurs immediately after the lecture on ordered models. The informal pre-class question I assign students is: "What happens when you estimate an ordered logit model to recover x's effect when y's true measurement level is nominal?" (Answer: $\widehat{\beta}_x$ will be biased.)

Using an app to set up the simulations removes the need for students to be aware of the simulation code—or to program in R *at all, for non-*R *courses. Instead, students can focus on the pedagogically salient points.*

Students navigate to a URL in their web browser to access the app in figure 1. All of the simulation settings appear in the "Main" tab's left sidebar. The settings are arranged broadly so that students make decisions about the simulations the same way they would if writing the actual R code. They decide how many observations per sampled dataset and how many samples to pull, what the data-generating process (DGP) will be in truth, and which model to estimate in an effort to recover x's effect on y. After they make these choices, students click the "Run Simulation" button.

A progress bar will appear as the simulations run. Once they are completed, new output appears in the main panel on the right. The app first reminds students whether y's DGP is ordered or nominal in truth and whether the reported model results are from an ordered or multinomial model, based on their selections in the left sidebar. Next, the app displays the expressions for the true DGP, set in LaTeX font. Finally, it displays the simulation results. Each column in the table represents an estimate out of the model; each row represents a quantity pertinent to the simulations.

If students are familiar with interpreting Monte Carlo simulation output, the table's layout makes the estimates' bias easier for students to see that a given parameter's true value (first row) does not fall within the 95% percentile-based confidence interval of its corresponding estimate (third and fourth rows). Students also can navigate to the "What Should I See?" tab for an explanation of how the estimates will be affected, given the selected DGP and model type, plus information about where to look in the simulation results table.

The app also provides opportunities to connect to previous lecture content via the other two tabs, all without forcing students to wrestle with the R code themselves. One tab displays a table with the regression results from every simulation draw ("Raw Simulation Output") to concretize from where the contents of the "Main" tab's result table originate. The other tab displays a histogram of each estimate's simulated values to reinforce the idea of sampling distributions ("Estimates: Distribution Plots").

CONCLUSION

This article suggests that instructors consider using interactive Shiny apps to better teach econometric models to their students. Shiny's interactivity makes this possible by making students more active participants in the learning process, by targeting similar concepts in different contexts, and by prompting students to reflect more frequently on what they see. I further expand on these various ideas, among others, in my monograph for the Cambridge Elements series (Metzger 2021). On the whole, I emphasize thinking about ways in which we can leverage the dynamicism from interactive apps to teach more effectively in methods courses, where there are many topics to cover in a brief amount of time. ■

NOTES

1. Portions of this article draw on Metzger's (2021) *Using Shiny to Teach Econometric Methods* and are reprinted with permission. All of the apps mentioned in the article are available on GitHub (www.github.com/MetzgerSK) in either the shinyElement or shinyAdvReg repository.
2. However, apps might be written specifically to enhance R's functionality for R users. shinystan is one such example.
3. See https://mc-stan.org/shinystan.
4. See http://experiments.berkeley.edu.
5. See https://jlsumner.shinyapps.io/electoralrules.
6. See https://cat.wfu.edu/resources/tools/estimator2.
7. The political science literature on teaching methods emphasizes similar themes (e.g., King and Sen 2013). Many other articles provide examples of concrete activities that may help to reach these pedagogical ends.
8. Deslauriers et al. (2019, 19251) also show that, despite performing better on tests, students *feel* like they learn less from active-learning methods, due "in part [to] the increased cognitive effort required during active learning."
9. For more on the dangers of minimal instructor guidance to students, see Kirschner, Sweller, and Clark (2006).
10. On a more cautionary note, Hancock and Rummerfield (2020) find that undergraduate students who first perform hands-on simulation activities gain more from subsequent computer-based simulations, suggesting that simulations must be properly introduced as a concept in their own right for some students to reap the full pedagogical benefits.
11. Doi et al. (2016) discuss using Shiny apps to teach various statistical concepts other than estimators' properties. Fawcett (2018, 2) discusses using Shiny apps in another pedagogical context: "…to help facilitate student interaction with methods from recently published papers in…extreme value theory and applications."
12. For users interested in modifying either of the simulation apps, I also discuss the R code's general structure.

REFERENCES

Bacon, Donald R., and Kim A. Stewart. 2006. "How Fast Do Students Forget What They Learn in Consumer Behavior? A Longitudinal Study." *Journal of Marketing Education* 28 (3): 181–92.

Bahls, Patrick. 2012. *Student Writing in the Quantitative Disciplines: A Guide for College Faculty.* San Francisco: Jossey-Bass Publishers.

Bailey, Michael A. 2019. "Teaching Statistics: Going from Scary, Boring, and Useless to, Well, Something Better." *PS: Political Science & Politics* 52 (2): 367–70.

Bean, John C. 2011. *Engaging Ideas: The Professor's Guide to Integrating Writing, Critical Thinking, and Active Learning in the Classroom.* 2nd ed. San Francisco: Jossey-Bass Publishers.

https://doi.org/10.1017/S1049096521001141 Published online by Cambridge University Press

Bekkerman, Anton. 2015. "The Role of Simulations in Econometrics Pedagogy." *Wiley Interdisciplinary Reviews: Computational Statistics* 7 (2): 160–65.

Biggs, John, and Catherine Tang. 2011. *Teaching for Quality Learning at University*. 4th ed. Maidenhead, UK: Open University Press.

Carsey, Thomas M., and Jeffrey J. Harden. 2013. *Monte Carlo Simulation and Resampling Methods for Social Science*. Los Angeles: SAGE Publications.

Cobb, George. 2015. "Mere Renovation Is Too Little Too Late: We Need to Rethink Our Undergraduate Curriculum from the Ground Up." *The American Statistician* 69 (4): 266–82.

Deslauriers, Louis, Logan S. McCarty, Kelly Miller, Kristina Callaghan, and Greg Kestin. 2019. "Measuring Actual Learning Versus Feeling of Learning in Response to Being Actively Engaged in the Classroom." *Proceedings of the National Academy of Sciences* 116 (39): 19251–57.

Doi, Jimmy, Gail Potter, Jimmy Wong, Irvin Alcaraz, and Peter Chi. 2016. "Web Application Teaching Tools for Statistics Using R and Shiny." *Technology Innovations in Statistics Education* 9 (1). https://escholarship.org/uc/item/00d4q8cp.

Fawcett, Lee. 2018. "Using Interactive Shiny Applications to Facilitate Research-Informed Learning and Teaching." *Journal of Statistics Education* 26 (1): 2–16.

Freeman, Scott, Sarah L. Eddy, Miles McDonough, Michelle K. Smith, Nnadozie Okoroafor, Hannah Jordt, and Mary Pat Wenderoth. 2014. "Active Learning Increases Student Performance in Science, Engineering, and Mathematics." *Proceedings of the National Academy of Sciences* 111 (23): 8410–15.

Groth, Randall E. 2013. *Teaching Mathematics in Grades 6–12: Developing Research-Based Instructional Practices*. Los Angeles: SAGE Publications.

Hancock, Stacey A., and Wendy Rummerfield. 2020. "Simulation Methods for Teaching Sampling Distributions: Should Hands-On Activities Precede the Computer?" *Journal of Statistics Education* 28 (1): 9–17.

Huber, Mary Taylor, Pat Hutchings, and Richard Gale. 2005. "Integrative Learning for Liberal Education." *Peer Review* 7 (3/4). www.aacu.org/publications-research/periodicals/integrative-learning-liberal-education.

King, Gary, and Maya Sen. 2013. "How Social Science Research Can Improve Teaching." *PS: Political Science & Politics* 46 (3): 621–29.

Kirschner, Paul A., John Sweller, and Richard E. Clark. 2006. "Why Minimal Guidance During Instruction Does Not Work: An Analysis of the Failure of Constructivist, Discovery, Problem-Based, Experiential, and Inquiry-Based Teaching." *Educational Psychologist* 41 (2): 75–86.

Menary, Richard. 2007. "Writing as Thinking." *Language Sciences* 29 (5): 621–32.

Metzger, Shawna K. 2021. *Using Shiny to Teach Econometric Models*. Cambridge: Cambridge University Press.

Prince, Michael. 2004. "Does Active Learning Work? A Review of the Research." *Journal of Engineering Education* 93 (3): 223–31.

Ruhl, Kathy L., Charles A. Hughes, and Patrick J. Schloss. 1987. "Using the Pause Procedure to Enhance Lecture Recall." *Teacher Education and Special Education* 10 (1): 14–18.

Yancey, Kathleen Blake. 2009. "Reflection and Electronic Portfolios: Inventing the Self and Reinventing the University." In *Electronic Portfolios 2.0: Emergent Research on Implementation and Impact*, eds. Darren Cambridge, Barbara L. Cambridge, and Kathleen Blake Yancey, 5–16. Sterling, VA: Stylus Publishing.

https://doi.org/10.1017/S1049096521001141 Published online by Cambridge University Press