The Software Behind the Stats: A Student Exploration of Software Trends Across Disciplines

Elizabeth Upton*
Department of Mathematics and Statistics, Williams College

Xizhen Cai
Department of Mathematics and Statistics, Williams College

Pamela Jakiela
Department of Economics, Williams College; CGD; IZA; BREAD; and J-PAL

Owen Ozier
Department of Economics, Williams College; IZA; BREAD; and J-PAL

Shyam Raman
Department of Economics, Williams College

January 27, 2026

Abstract

This paper presents a student activity designed to explore the use of software and reproducibility practices in academic research across selected journals in economics, political science, and statistics. The activity was designed to deepen students' understanding of reproducible research workflows and increase their awareness of disciplinary norms in quantitative scholarship. Students reviewed replication files associated with published articles from major journals and repositories, using a survey to document software use. Combined with web-scraped metadata, the resulting dataset spans more than 10,000 papers and reveals clear patterns within the journals examined: Stata remains dominant in economics, R is increasingly common in political science, and R is the standard in statistics, with a growing number

*The authors are grateful to Richard De Veaux for helpful comments on an early draft; to David Zimmerman for crucial early stage work in the project and for providing his support throughout; and to Isabel Beckrich and Jonathan Hartanto for excellent research assistance.

of articles using multiple software platforms within a single manuscript. These results provide educators with a data-driven view of software practices in contemporary quantitative research, helping to inform decisions about which tools to emphasize in the classroom. To assess whether the student activity met its pedagogical goals, we summarize student feedback, which indicates increased understanding of academic workflows and greater awareness of software diversity. We provide all instructional materials, data, and code to support adoption across course levels and disciplines, and conclude with suggestions for follow-up assignments.

Keywords: Stata, R, Python, economics, political science, reproducibility

1 Introduction

A range of academic departments, including statistics, economics, and political science, offer courses in statistical and econometric methods. The teaching is not entirely theoretical; recent work has documented and encouraged a pedagogical shift towards empirical examples (Angrist and Pischke, 2017; Carver et al., 2016). Classes on these topics now regularly include exercises that require the use of widely-used statistical software packages. Which programming language or statistical package, however, seems to vary across disciplines and is constantly changing over time (Bednarowska-Michaiel and Uprichard, 2025).

As educators, we make a conscious decision about what software tools we will use in our courses (Schwab-McCoy et al., 2021; Siegfried et al., 2021). This choice is often based on our own experiences, backgrounds, and the resources available to us. A pragmatic consideration should also be the current state of the software landscape—that is, an understanding of what software is being used in different disciplines. For example, economics departments teach in Stata at least in part because published economic research uses Stata—or at least, conventional wisdom suggests that this is the case, though data on this front is limited.

From the student perspective, understanding the software landscape is also valuable. Students make decisions about which courses to take and how much effort to invest in learning a particular software tool or programming language. While it's easier to run pre-written code and produce results, developing fluency in a tool requires time and motivation. The classic question—"why do I need to know this?"—isn't about whether software is useful, but rather which software is worth learning and why. A clearer picture of disciplinary software norms can help students better appreciate the relevance of what they're learning, and give them more confidence in navigating their own analytical work.

While software preferences shift, one fundamental principle remains constant: reproducibility (i.e. obtaining consistent results using the same input data (Natl. Acad. Sci. Eng. Med, 2019)) is essential for rigorous research (Munafò et al., 2017; Gertler et al., 2018; Vilhuber et al., 2022; Wrobel et al., 2024; Dogucu, 2024). Yet, while we tell our students that their code should be shareable and readable, they may not understand how reproducibility pipelines are operationalized in practice. Students may initially be unaware of the idea of reproducibility in data analyses (Ostblom and Timbers, 2022); even in classes that use data, a recent survey showed that relatively few classes introduce students to common ways of transparently sharing data and code (Underwood et al., 2024).

Additionally, we often do not emphasize how reproducible workflows can be leveraged by students in their own research. That is, reproducibility is not just about making their own work transparent for others—its existence enables students to replicate findings from a paper, learn how to practically apply methods, and build upon existing research with confidence. Our courses, then, should not only teach students how to use the statistical tools prevalent in their field but also foster an understanding of reproducibility as a core research practice. At a low cost, we could also expose students to the tools used in other disciplines, emphasizing how the skills developed in one software package can transfer to another—and how these tools, in combination, support replicable research.

To this end, we designed a short activity for students that provides hands-on practice in accessing and interacting with existing replication packages, an essential first step in fostering reproducible research skills. The results of the activity, combined with web-scraped metadata, form a dataset describing software use in over 10,000 academic publications across statistics, economics, and political science journals. This dataset offers insight into the current landscape of statistical software usage, information that is valuable for students interested in quantitative fields of study and for instructors making decisions about which software tools to use in their classes. Thus, the contributions of this paper include (1) the introduction of a student activity, easily adaptable to a variety of classroom settings, that exposes students to reproducibility in academia; (2) the use of the resulting dataset to identify software trends that can inform learning and teaching practices; and (3) an outline of several potential extensions that build on or make use of the dataset, including student activities focused on reproducibility verification, documentation quality, and software comparison.

The remainder of the manuscript is organized as follows. We first provide historical context and review prior work on trends in software use in research and teaching. We then describe the data sources and the student activity used to construct the dataset, present findings derived from the dataset, and conclude with a discussion of student outcomes and proposed extensions. This study was reviewed by the Williams College Institutional Review Board and determined to be exempt from IRB review, as it presents no risk to participants and relies exclusively on publicly available data. To support transparency and reproducibility, all data and analysis code used in this study are available in an Open Science Framework (OSF) repository[1].

2 Background

To contextualize the changing software landscape, approximate dates of the releases of some recent software packages and languages are provided in Table 1.[2] In addition to statistical packages and interactive environments, the table includes two programming languages—Python and Julia—since some research and teaching may use them.[3] However, as statistical software options have changed dramatically over recent decades (as have computing options more generally), there has been relatively little comprehensive documentation of the trends in statistical software use in academic publications. We discuss some recent reviews below, though they each have limitations.

Fišar et al. (2024), for example, report on relative software popularity for articles published in *Management Science* between 2019 and 2023. They find that Stata is the most commonly used (60% of articles), followed by R (19%), MATLAB (18%), SAS (13%), and Python (11%), with other analytical software being used much less frequently. This provides a recent picture, but at one journal and for only a few years.

Vilhuber (2020) documents trends from 2010 to 2019, restricting attention to journals published by the American Economic Association. He shows that Stata is present in the majority of replication packages for those journals in every year, with MATLAB in second place, and a changing cast of others in a relatively distant third (SAS in some years, R or other software in others).

Christodoulou (2015), in an analysis published by Stata Press, provides some earlier patterns of software use. He describes a literature search procedure for "citations of notable statistical software: GenStat, GLIM, MATLAB, Minitab, NLOGIT, Octave, RATS, SAS, SHAZAM, SPSS, Stata, Systat, and TSP." The analysis excludes the ubiquitous Microsoft Excel because "it is not marketed as a statistical software," but makes a more particular note regarding the approach's limitation in relation to R: "The most notable omission is the R-project due to the commonality in its keyword (that is, the single letter 'R')." (p.101) That analysis shows that as far back as 2000,

[1] `https://osf.io/72ajq/overview?view_only=fdc707fcf33848678a4583ae90b36514`

[2] Many more entries could be included in this table. The present paper is not intended to be a comprehensive history, but this table serves to situate current popular options in some historical context. Renfro (2004) describes the history of related software in much more detail, including the development of TSP on Univac machines, the transformation of MicroTSP into EViews, and so on.

[3] Older languages such as FORTRAN and C are still used directly in some research, but we omit them from the table as we are not trying to establish a complete history of mathematical computing.

Table 1: Selected statistical computing release timeline.

Year	**Software**	**Citation**
1968	SPSS	(Wilson and Lorenz, 2015)
1972	Minitab	(Schissler et al., 2014)
1976	SAS	(Rodriguez, 2011)
1976	S	(Chambers, 2020)
1980	RATS	(Renfro, 2004)
1981	LimDep	(Renfro, 2004)
1984	MATLAB	(Chonacky and Winch, 2005)
1984	GAUSS	(Anderson, 1992)
1985	Stata	(Renfro, 2004)
1991	S-Plus	(Hallman, 1993)
1991	Python	(Van Rossum and Drake, 2003)
2000	R	(Chambers, 2020)
2012	Julia	(Perkel et al., 2019)

in each of the research areas described as "Business," "Accounting," and "Finance," the leading two statistical software packages were consistently SAS and SPSS. By 2013, in all three areas, SAS had dropped out of the top two spots, and Stata was one of the top two, along with MATLAB or SPSS.[4]

We found comparatively little research examining trends in software use over time in undergraduate teaching. Instead, much of the existing literature focuses on qualitative considerations such as the relative cost or ease of use of different software tools. For example, Conaway et al. (2018) note that "students do not seem to find R as easy to learn as Stata," while also emphasizing that R may be the most practical option for students without access to proprietary software.

In the absence of systematic studies documenting trends, textbooks offer an alternative lens for understanding the instructional software landscape. Some econometrics texts emphasize a single language—for instance, Angrist and Pischke (2009) includes Stata code throughout—while others reflect shifts over time. The econometrics text by Wooldridge (2016), for example, long provided datasets in formats such as Stata, Eviews, Minitab, and Excel, but began including R datasets in its sixth edition (2016). More recent texts often adopt a multi-language approach: Bailey

[4]Relative citation counts from 1999 to 2013 are shown in that study's Figures 13.1, 13.2, and 13.3. The graphs include SPSS, Stata, SAS, MATLAB, RATS, EViews, S-Plus, and Limdep.

(2019) includes both R and Stata examples, explicitly discussing trade-offs between usability and cost. In the field of statistics education, one can find teaching examples in textbooks that use R, Python, SPSS, Stata and more (e.g., Ferrall, 1995; Tucker et al., 2023; De Veaux et al., 2005). The data science text by Baumer et al. (2017), utilized across disciplines, uses R throughout, while acknowledging Python as a widely used alternative and noting the lack of consensus around a single "best" language.

There do appear to be disciplinary differences in software choice. To illustrate this, we conducted keyword searches for articles published over the past 30 years (1995–2024) in journals focused on teaching within each of three disciplines (Figure 1). In both the *Journal of Economic Education* and the *Journal of Political Science Education*, more articles explicitly mention Stata than mention either Python or R.[5] However, in the *Journal of Statistics and Data Science Education*,[6] the pattern is very different: there are more articles mentioning R or Python than there are mentioning Stata. A recent article by Best and Mallinson (2024) describes a shift over the last two decades in quantitative political science away from "proprietary software like SPSS, Minitab, SAS, or STATA" towards a "more diverse" set of tools including "R and Python." Our search in pedagogy journals is limited, however: we examine only a single journal per discipline, which may or may not be representative of the entire landscape. Furthermore, with only 43 total articles mentioning any of these types of software in the *Journal of Economic Education* and only 23 such articles in the *Journal of Political Science Education*, we cannot say much about trends from this dataset.

3 Data Collection and Student Activity

Our dataset documenting software usage across statistics, economics and political science, arises from a two-pronged approach. First, we systematically scrape large online repositories for metadata. This provides a partial picture of trends in current research. Second, via the student activity, we ask students to visit a sample of articles and repositories to see for themselves what files are present. The two approaches complement one another: the systematic approach produces far more

[5] For Stata and Python, we search for a single keyword. For R, we search for "Rstudio" or "R programming" or "R code" rather than R, aware this may under count examples of R, but we do so because it is easier to search for a string more than one character long. The comparison of relative rates across journals is still informative even if the levels suffer from differential biases.

[6] Historically, this journal primarily focused on statistics. In 2021, it incorporated "Data Science" into its name to reflect the growing importance of the field.

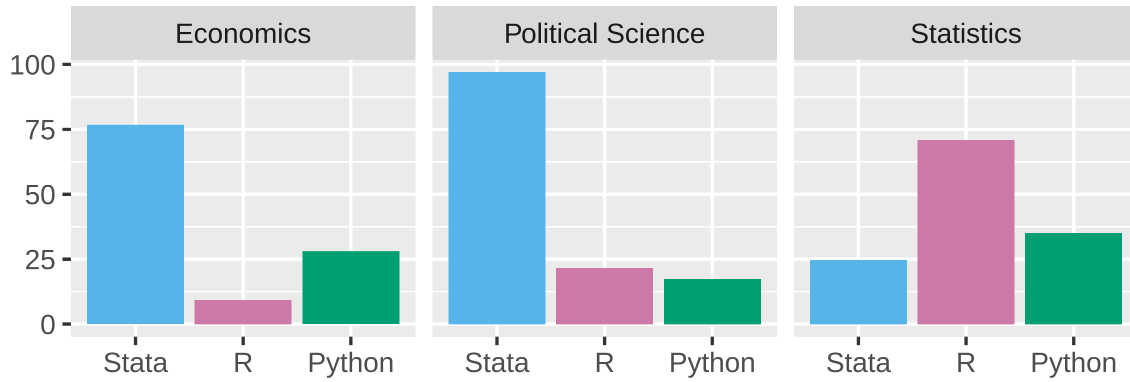


Figure 1: Mentions of software in three journals. Percents are reported among articles mentioning any of "Stata" or "Python" or "Rstudio" or "R programming" or "R code" in the *Journal of Economic Education*, the *Journal of Political Science Education*, and the *Journal of Statistics and Data Science Education*, from 1995 through 2024. Sample sizes: 43 in economics; 23 in political science; 154 in statistics and data science.

data more quickly than the manual approach; but manual searches are particularly useful for any journal that does not have a centralized online location for replication files (the data, code, and documentation necessary to reproduce the main analyses and results reported in a paper), and even when the systematic approach is possible, manual cross-checking can validate it.

3.1 Data sources

The five major sources of data include three specific journals or publishers: the *Journal of the American Statistical Association* (JASA), the *American Economic Review* (AER), and the *American Political Science Review* (APSR). These are each flagship journals of the scholarly association associated with each field in the United States. Two other data sources are the the repository for the American Economic Association (AEA) hosted by the Inter-university Consortium for Political and Social Research (ICPSR) and the Harvard Dataverse—which includes analysis repositories from multiple journals in multiple disciplines, and for which we focus on economics and political science journals.[7] Across the Harvard Dataverse and the AEA repositories at the ICPSR, our automated data collection includes all the so-called "Big 3" journals in political science, as well as three of the so-called "Top 5" journals in economics. Details on the sample sizes, journals, and

[7]The AEA collection at ICPSR has been studied in other ways as well; for example, by Li et al. (2024). Underwood et al. (2024) have also highlighted the Dataverse as a leading example of research accessibility resources from which students might benefit.

years covered by these sources are provided in Online Appendix Table A1. Note that the data were scraped in the fall of 2024, affecting 2024 sample sizes for the automated data collections.

Sources were chosen based on our familiarity with them as scholars, their topical relevance and comprehensiveness, and the ease of accessing information. For example, we explored including articles from the Institute of Mathematical Statistics (IMS)(i.e. *Annals of Statistics* and *Annals of Applied Statistics*) but found it difficult to get a succinct list of DOI stubs and for students to be able to reliably navigate to article content. These choices necessarily limit the scope of our conclusions: our findings should not be generalized to all of statistics, economics, or political science. We return to this limitation in the discussion, where it becomes a productive point of reflection for students.

As mentioned, there are limitations to the scrapable metadata that can be overcome by our student activity. For JASA, the manual student approach produces more current data than is otherwise available.[8] We also note that the website of the AEA, in its Terms of Use, forbids the use of "scraping programs" to harvest content directly from its site. For the Dataverse, the manual student-driven approach allows us to cross-check our use of metadata, and allows us to know what is inside zipped archives that are otherwise not transparent via metadata. Furthermore, conditioning on the existence of an online repository does not allow us to answer the question, "What fraction of papers are accompanied by replication files?" Repository scraping necessarily restricts attention to manuscripts that already have repositories; papers without replication materials are never observed and therefore cannot be counted. In contrast, student inspection at the journal level allows us to estimate the proportion of all manuscripts within a journal that include replication files, and also captures papers that link to external data or code sources outside standard repositories.

Both the automated analysis of file types and our student-driven activity examining files themselves overcome the limitations of purely keyword-based analysis mentioned in previous work (Christodoulou, 2015).

[8]There is a set of 109 GitHub repositories associated with JASA from 2018 to 2021, but we do not graph the trends there in detail both because it is not clear whether it is a representative sample from that time period and because that series does not extend to the present. We do note that 83% of those repositories use R; 16% use MATLAB; 17% use C or C++; 3% use Julia, 2% use Python, and 1% use each of Stata, Stan, and FORTRAN.

3.2 Student Assignment Overview

For the manual approach to data collection, we developed a student assignment in which participants visited the webpages of journal articles. This gave students an opportunity to engage directly with replication materials as they appear in practice. Each student reviewed a small number of articles—across economics, political science, and statistics—identifying the software used. This familiarized students with replication practices in three fields, while enabling us to build new datasets and to validate our automated analysis.

Students participating in the activity received an assignment (see supplementary materials) with the prompt:

"The goal is to see how researchers working in different disciplines share materials for replication of empirical statistical analyses (and what software they use). We will visualize the results once everyone has done their small piece of data collection. For your part, this means: (a) visiting a website relating to a published paper, (b) exploring it in a structured way, (c) putting what you find into an online survey, and (d) recording in our course website that you have done so."

The students were instructed on how to access their assigned websites, either via a provided URL or a shortened digital object identifier (DOI). Their instructions included guidance on how to identify relevant files within replication packages and repositories: we gave them a list of file types to look out for (e.g. **.R, .Rda, .do, .py**) and instructions for how to open compressed folders. If no replication files were available, students were asked to skim the article for terms indicating the software used or alternative locations where the files might be found.

Once students completed their search, they recorded their findings in a structured Qualtrics survey. The survey began with general information collection:

1. Provide the URL or DOI stub associated with this survey entry
2. Before you answer detailed questions about this entry, was there anything unusual that you would like to comment on?
3. Were you able to find any replication files associated with this DOI? (choice: Yes/No)

If the student selected yes on question 3, they were presented with a series of follow up questions, such as:

- Were you able to see any R code files (ending in .R, .Rda, .Rmd)[9] with the replication materials?

with response options Yes/No. If the student selected no on question 3, they were asked if they found any description of the software used in the article. An answer of no would bring them to the final survey question and an answer of yes would ask them to choose the software mentioned and provided a textbox for them to enter the exact wording used by the authors in the article. The final survey question asked students whether they considered categorizing the files as a clear or ambiguous task. To read the complete Qualtrics survey blueprint, see the supplementary materials.

To ensure familiarity with the process before collecting data, we prepared two warm-up exercises for the students to complete. We requested that they complete the survey based on fictional entries, which we describe for them in detail. We also provided the students with a list of example DOIs whose content was known. In preparation for their own data collection, they could visit these links, explore the files, and verify the types of files they identified against the known content.

3.3 Student assignment protocol and sample sizes

Five professors at the authors' college incorporated the assignment into their courses, three from economics and two from statistics. Before introducing the activity in classes, we conducted a test run with a small group of research and teaching assistants. These students also provided feedback on the average time required to complete the task. All agreed that to check ten articles or replication packages via DOI took less than two hours; for all but one student it took less than one hour. We confirmed the duration by asking student participants how long the larger-scale activity took them. A typical duration reported for ten such checks was one hour: 24.5% said it took between 20 and 40 minutes; the plurality (43.9%) said it took between 40 minutes and 1 hour; 28.6% said it took between 1 and 2 hours, and 3.1% said it took more than 2 hours.

In total, 158 students across seven separate classes and sections were assigned to review DOIs. Each student was assigned four DOIs from JASA and six DOIs from economics and political science. To assign DOIs, we created a master file with student IDs and randomly allocated 10 DOI stubs to each student. Each article was initially assigned to two students for review, though never two students in the same class. This yielded a total of 790 unique DOIs. For example, of 316 initially

[9]Students were asked about R, Stata, MATLAB, C/C++, SAS, Python, SPSS, and 'Other' file types

assigned JASA DOIs, 214 received at least one student visit and yielded a usable classification outcome. Among these, 131 DOIs were visited by more than one student and classified consistently (across every question), 9 were visited by at least three students with a majority agreement on classification, and 74 were visited by exactly one student.[10] Of these 214 classified DOIs, 69 did not have any files to report or describe.[11] We use the remaining 145 JASA cases as the basis for analysis in Figure 8.

To cross-check automated collection (and to verify student data quality), we assigned a small number of automatically classifed Dataverse entries to students. Of 68 repositories of this kind, we checked MATLAB, Stata, and R status across the automatic and student classification, for 204 comparisons. There were 201 agreements and 3 disagreements. In one case, the students correctly noticed a filetype that our initial automated procedure had incorrectly omitted. We revised the automatic procedure to capture this. In the other two cases, the students reported seeing both Stata and R when in fact there was only R. The student error rate is thus roughly 1 percent.

Interested students who had asked for more work of this kind were hired as research assistants. They checked some of the DOIs that previous students had missed or that had been inconsistently classified previously, and they carried out additional comprehensive data collection for the most recent years of three journals.

4 Findings

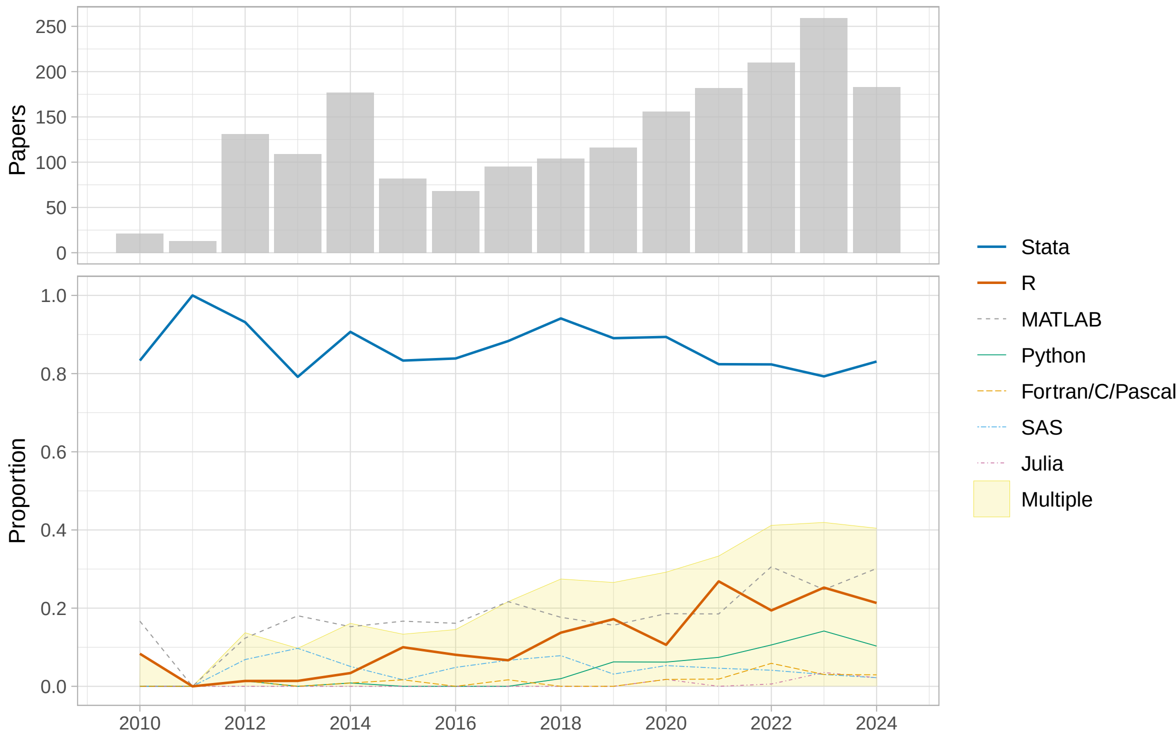
Figures 2 to 5 show statistical computing trends over time, as represented by our sample of published papers and replication packages from economics, political science, and statistics.

In the economics journals, Stata is dominant, appearing in 80% (or more) of replication packages (Figures 2 and 3). Moreover, there is no evidence that use of Stata is declining over time in economics. For much of the last fifteen years, MATLAB has been the second most widely used statistical software, but in recent years R has been catching up; in 2023 and 2024, both MATLAB and R were included in more than 20% of replication packages in both the Harvard Dataverse

[10] Fourteen DOIs were not visited by any students, as not all students completed the assignment. An additional 88 DOIs were visited by an even number of students (almost always two) who disagreed on at least one survey question. We excluded these cases from the analysis. Some DOIs were visited by more than two students due to test runs or students requesting additional work.

[11] This occurred when the assigned DOI corresponded to an editor's note, theoretical article, or review article, or when no replication files were available.

Figure 2: Software in economics journal publications posted to Harvard Dataverse



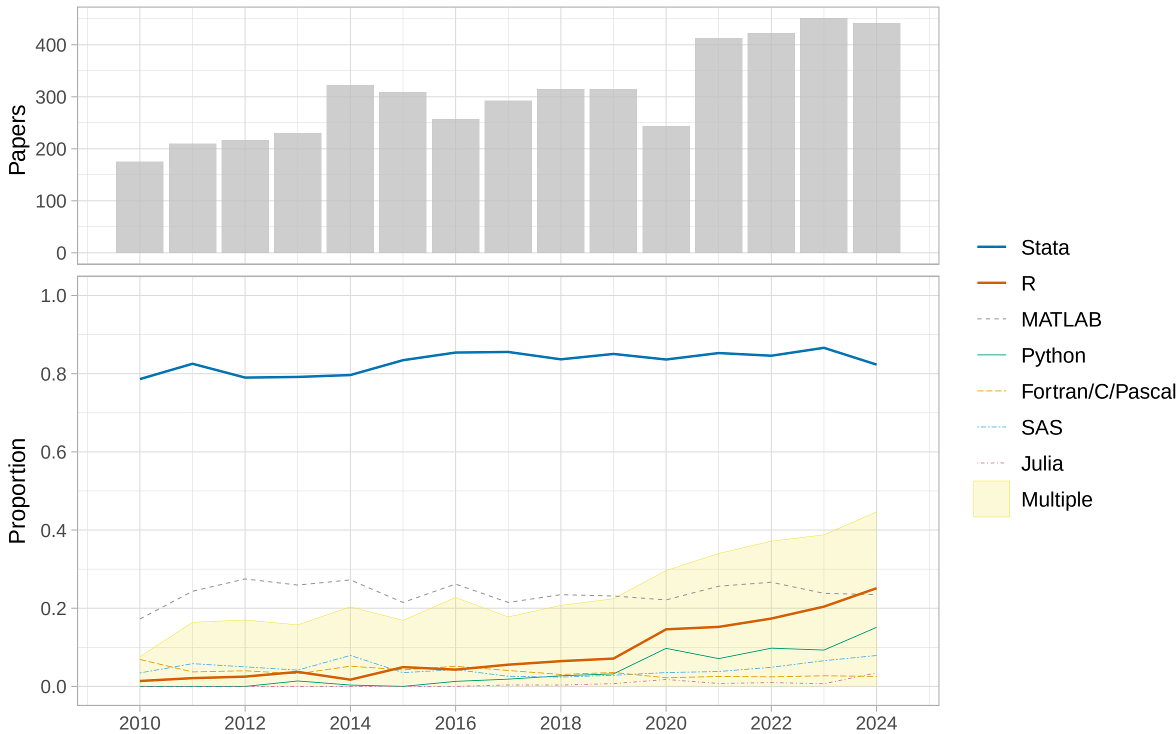
Note: Figure presents trends in software used by replication packages from $N = 1{,}906$ papers published in economics journals. Data were collected via automated file categorization and validated via student exploration. Upper panel reports the number of articles examined in each year; lower panel reports proportions (scale from 0.0 to 1.0).

Source: Data sourced from Harvard Dataverse for the following journals: *Journal of Political Economy*, *Journal of Political Economy: Macroeconomics*, *Journal of Political Economy: Microeconomics*, *Quarterly Journal of Economics*, and *Review of Economics and Statistics*.

and AEA/ICPSR data sets. Python is also increasing in popularity: while almost no replication packages included Python files prior to 2016, Python is now present in more than 10% of replication packages (and, again, this pattern is apparent in both the Dataverse and AEA/ICPSR data sets).

As these numbers suggest, within economics there has been a general trend toward the use of multiple statistical software tools over the last fifteen years. In 2010, the proportion of replication packages that included two different types of statistical software was below 10% in both the Dataverse and AEA/ICPSR samples. By 2024, more than 40% of replication packages in both data sets included at least two different types of statistical software. Thus, while Stata remains dominant in economics, the evidence suggests it is increasingly the case that economists use Stata in conjunction with other statistical computing tools such as MATLAB, R, and Python.

Figure 3: Software in replication packages from the AEA ICPSR repository.



Note: Figure presents trends in software used by replication packages from $N = 4{,}617$ papers published in economics journals. Data were collected via automated file categorization. Upper panel reports the number of articles examined in each year; lower panel reports proportions (scale from 0.0 to 1.0).

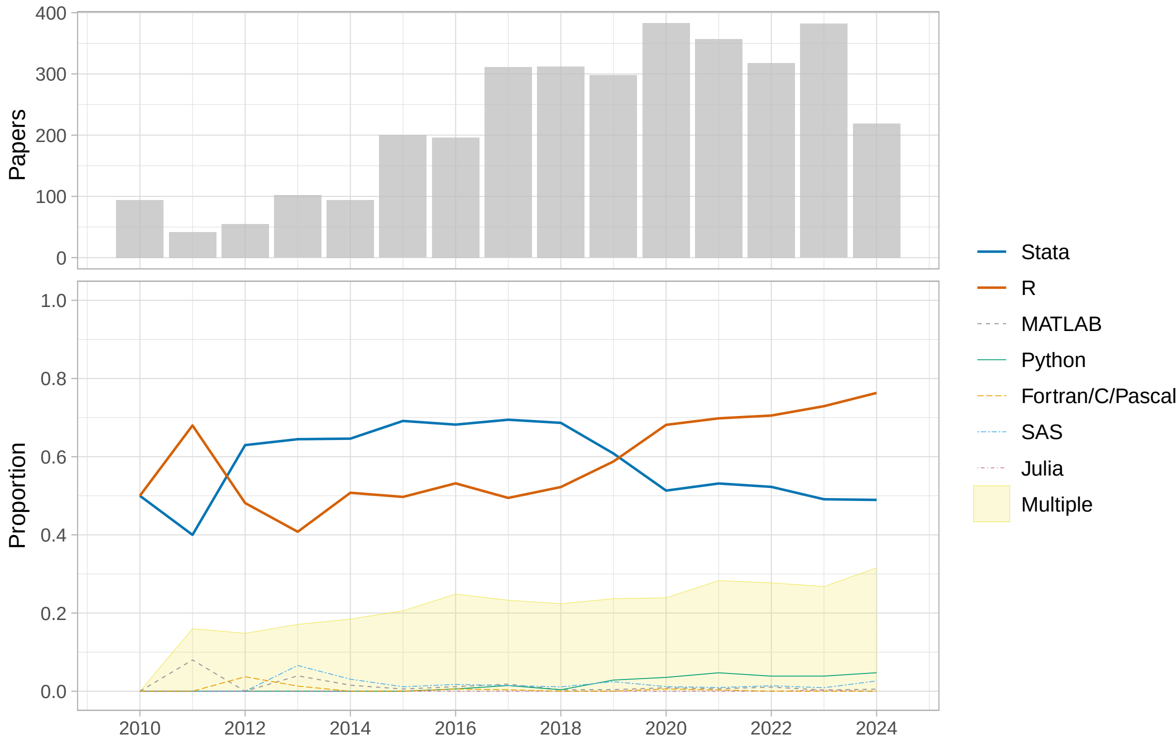
Source: Data sourced from the AEA's ICPSR repository for the following journals: *American Economic Journal: Applied Economics*, *American Economic Journal: Economic Policy*, *American Economic Journal: Macroeconomics*, *American Economic Journal: Microeconomics*, *American Economic Review*, *Journal of Economic Literature*, and *Journal of Economic Perspectives*.

In the political science journals, we also observe a broad trend toward the use of multiple statistical computing tools (Figure 4). Both Stata and R are widely used within these journals, though R crossed an equal-popularity threshold in 2019 and appears to be trending in the positive direction, while Stata appears to be trending down slightly. However, Stata files are still present in approximately half of all replication packages from 2024, while R is now present in approximately 75%. While the frequency of replication packages containing multiple types of statistical software has increased over time, the trend is not quite as marked as in economics: by 2012, more than 15% of replication packages in our political science sample included at least two types of statistical software, and this increased to about 30% by 2024. In contrast to economics, there is little evidence of MATLAB or Python (or any other type of statistical software) gaining ground in political science.

While the broad trend in the economics and political science journals we reviewed appears to be toward greater use of multiple flavors of statistical software, there is little evidence of such a trend in the flagship statistics journal. Very few statistics articles (less than 1%) in JASA (out of hundreds we checked) report using Stata or include Stata in their replication package (Figure 5). Around 87% of JASA articles use R, with MATLAB coming in a distant second (12% of articles sampled) and Python third (8% of articles). There is some evidence that Python use is increasing over time, though overall adoption remains low. However, the evidence from statistics stands in stark contrast with what we observed in quantitative social science: fewer than 10% of JASA articles use more than one type of statistical software, and there is no evidence of the use of multiple types of software is increasing over time.

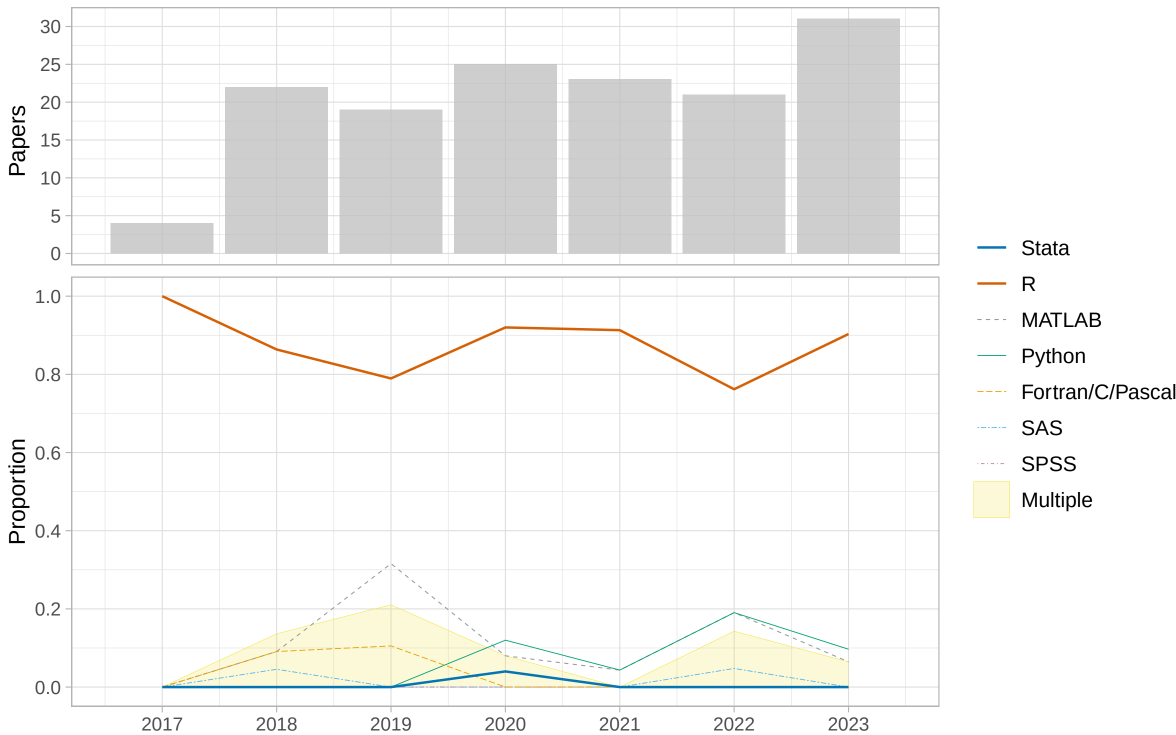
To explore the increasing use of multiple software platforms and languages, we visualized the intersections of the tools in a single journal from each discipline in Figures 6:8. The data shown comes from 2024 and was collected via student exploration. We see, for example, that 35 articles published by the *American Economic Review* use Stata, 14 use Stata and MATLAB, and 11 use Stata and R. In the *American Political Science Review*, R is used on its own in 53 articles while Stata is used alone in only 14 – but the use of R in conjunction with Stata is quite common, occurring in 26 different articles. As expected, R used alone is the most common pattern observed in JASA, occurring in 127 papers from 2024. R and Python were used together in 22 papers, while R was used with C/C++ in 20 papers. Thus, here again we see a marked difference between statistics – where R remains uniquely domiant – and quantitative economics and political science, where both R and Stata are common and we observe a clear trend toward combining different

Figure 4: Software in political science journal publications posted to Harvard Dataverse



Note: Figure presents trends in software used by replication packages from $N = 3{,}363$ papers published in political science journals. Data were collected via automated file categorization and validated via student exploration. Upper panel reports the number of articles examined in each year; lower panel reports proportions (scale from 0.0 to 1.0).
Source: Data sourced from Harvard Dataverse for the following journals: *American Journal of Political Science*, *American Political Science Review*, *British Journal of Political Science*, *Journal of Politics*, and *Political Analysis*.

Figure 5: Trends in software for Journal of the American Statistical Association (JASA) articles



Note: Figure presents trends in software used by replication packages from $N = 145$ papers published in JASA. Data were collected via student exploration. Upper panel reports the number of articles examined in each year; lower panel reports proportions (scale from 0.0 to 1.0).
Source: Student data collection from the JASA website.

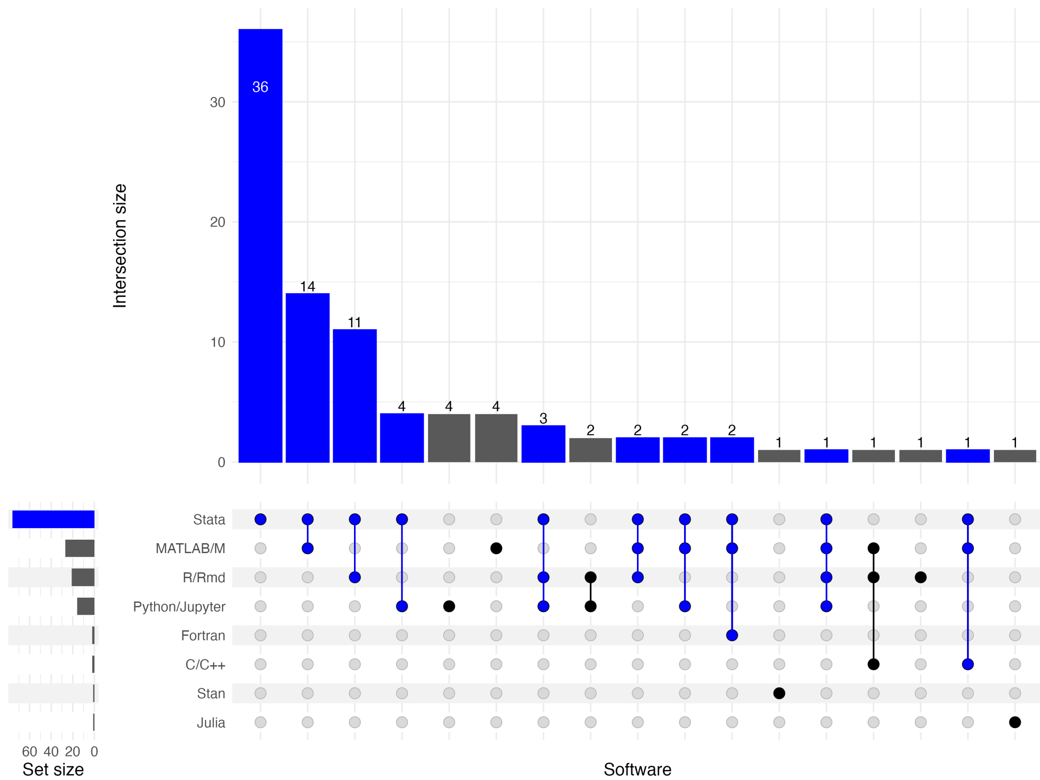


Figure 6: Software in 2024 issues of the American Economic Review. Restricted to 90 articles (82 percent) in which software could be categorized out of 110 articles in total.

types of software.

Many of these patterns align with what one might have expected (economists tend to use Stata; statisticians favor R), but it's satisfying to have data to support those assumptions. Other trends, such as R and MATLAB now being equally common among economists, were less obvious ex ante. Another clear takeaway is the number of manuscripts, across the board, that utilize multiple software platforms. This may reflect increased collaboration across academic disciplines, or the growing ease with which researchers can learn and adopt multiple tools, thanks to the proliferation of online tutorials, open-source communities, and more accessible training resources.[12]

Students also found something that our automated look at large repositories could not: the type of statistical software used was evident (either through replication code availability or through explicit text description) in 91 percent of articles examined in JASA, while this was true of only 71 percent of articles in the APSR; the AER fell between the two at 82 percent. This pattern may mainly reflect the types of articles in the journals: theoretical papers or Nobel lectures might not require new statistical analysis. There were also limitations to our approaches. For example, students were asked to systematically search for information regarding popular statistical software, but while less common software and packages were occasionally mentioned in student comments, we cannot be sure how frequently they are mentioned because we did not ask students to systematically search for them.[13]

5 Discussion

The student activity was introduced as part of a broader, campus-wide data collection effort, with students contributing to a shared dataset. Prior to implementation, instructors expressed some concern that students might view the task as busy work, particularly because it involved navigating journal folders rather than traditional problem solving. In practice, the opposite occurred. Students expressed excitement about participating in a collaborative data collection effort and reported little to no prior awareness that academic manuscripts are often accompanied by online

[12]It is also worth noting that some software packages rely on libraries implemented in other programming languages. For example, many R packages call underlying C or C++ code. Our process likely does not capture these dependencies, but this is a useful and important conversation to have with students.

[13]There were two student mentions of Gurobi, one of gretl, and one of Gephi, for example, but no student mentions of Minitab, LimDep, or EViews. Geographic analysis packages such as ArcGIS and QGIS are mentioned in a number of instances as well.

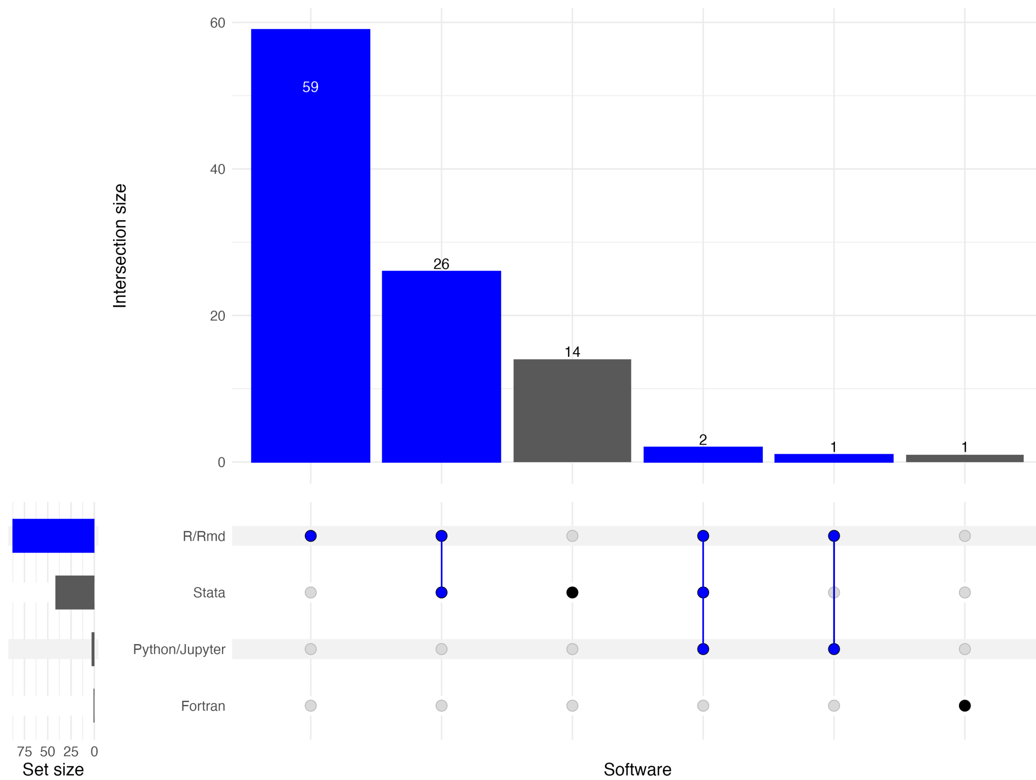


Figure 7: Software in 2024 issues of the American Political Science Review. Restricted to 103 articles (71 percent) in which software could be categorized out of 146 articles in total.

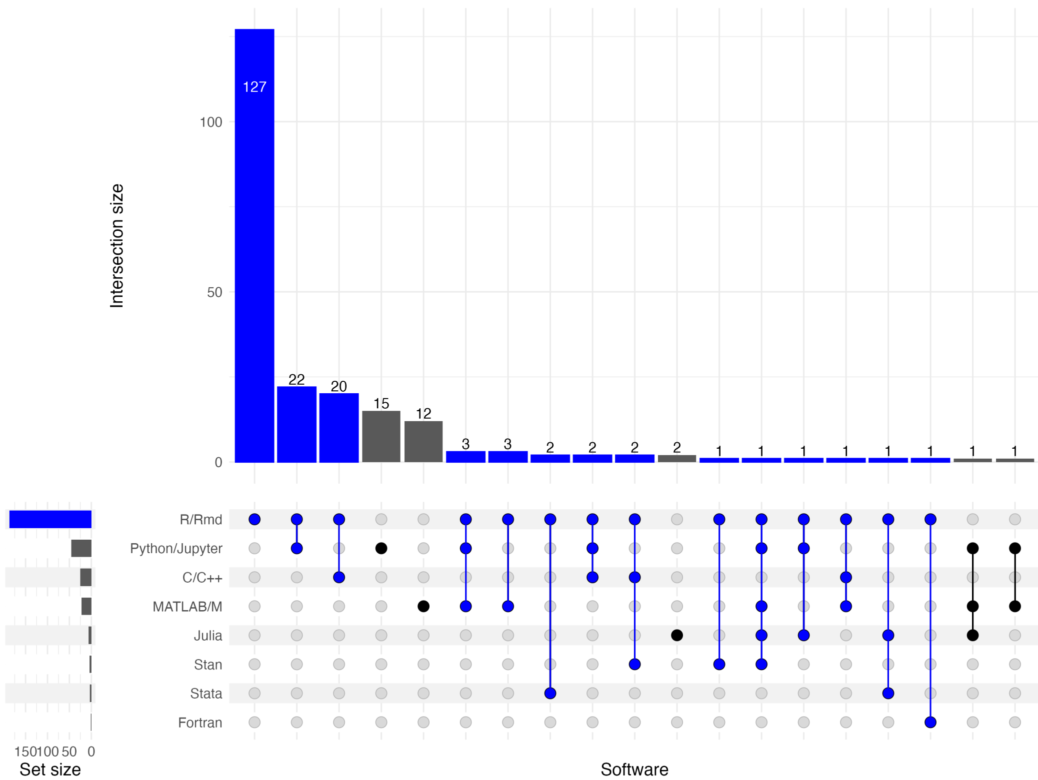


Figure 8: Software in 2024 issues of the Journal of the American Statistical Association. Restricted to 218 articles (91 percent) in which software could be categorized out of 240 articles in total.

repositories containing data, code, and documentation. Furthermore, in our experience, students found exploring journal websites and academic articles both fun and eye-opening. The comments left by the students on the survey provide evidence the activity was well-designed:

"I didn't encounter any issues during this task. All the DOIs that were assigned to me were easy to navigate. I was able to find the replication materials pretty quickly."

"Was confused at first how the websites work / where to find the data but figured it out and it became quick!"

General consensus was that the JASA articles were the most variable and difficult to navigate, but the students enjoyed browsing the articles and learned to differentiate between common sections - derivations, simulations, and data analyses. Similarly, the students often pointed out when a manuscript did not meet their reproducibility criteria:

"mentions github repository with more details but could not find link"

"there seems to at one point have been a replication package but it is currently temporarily unavailable"

Student engagement was evident in a senior-level statistics seminar, where there was audible cheering upon seeing that R was gaining popularity in political science. Meanwhile, some students questioned Python's relatively low usage in JASA articles. This prompted a conversation about the scope of our conclusions, including how journal focus and subfield concentration mean that JASA does not represent statistics as a whole. Anecdotally, students expressed an appreciation for having a better understanding of how academic research is organized, accessible, and presented across different fields.

We also shared these graphics with both an intermediate econometrics and an advanced microeconometrics course in the economics department. Sharing the data in econometrics courses served different purposes at different moments in the courses. It motivated training in Stata so that students could engage with the most common language in replication packages; it also motivated advanced assignments that took place in multiple languages (Stata and R) or that required translation (from Stata to Python or R) since our data show multilingual research environments to be increasingly common.

This activity described in section 3.2 is intended as a first step in introducing students to the reproducibility pipeline. To build on the initial data-collection phase, we have identified several follow-up activities that reinforce key learning objectives—particularly around reproducibility and software fluency. Some of these tasks we have implemented ourselves, while others serve as recommendations for future courses depending on the pedagogical goals:

1. Provide students with the full dataset collected and have them create their own data visualizations that summarize the current software landscape. The possibilities here are numerous, but the nature of the data (time series/categorical variables) makes this an interesting exercise in data visualization.

2. Ask students to pick a manuscript that includes replication files, replicate its primary analysis, and then translate the original code to another software environment. Comparing the two versions helps students appreciate both the similarities and subtleties in different programming languages.

3. Replicate and expand upon an existing data analysis from the collection. Students should identify and evaluate the authors' data-processing choices and note how they might do things differently. If the original analysis lacks explicit assumption checks (e.g., residual plots), students should perform and interpret them as part of this extension.

4. Have students assess the reproducibility of the provided code and documentation. Is the code well-commented and easy to follow? Does a README file clearly introduce the project and explain how to run everything? When present, do automated testing suites or example scripts provide additional evidence that the code behaves as intended? These questions help students see the real-world importance of sharing reproducible research.

5. For students particularly interested in the activity, offer research assistant opportunities in which they could gather more data. This could focus either on specific journals to create a more comprehensive picture for particular subfields or on disciplines we have not yet considered.

A key takeaway from this activity is the importance of being multilingual, however we recognize that limited class time (and student interest) can prevent a deep dive into multiple software environments in any single course. In most of our classes, software is primarily a means to carry out analyses and we cannot devote too much time, if any, to learning more than one tool. Still, even brief exposure to additional languages can be valuable: students may be proficient in Stata, for instance, yet they should realize that the coding principles they've learned will map onto R, Python, or other platforms. For introductory courses, perhaps a single class should focus on a single language. But students should know that there is more than one of them. Finally, by emphasizing reproducible workflows in this activity, we show students how accessible, well-documented code can be a resource that not only supports replication but can also help them quickly adopt and master new methodologies.

Even in statistics, we can see that though R is by far the most popular language, the use of multiple languages is not uncommon. The present manuscript is no exception. In writing this paper, we used both R and Stata for web scraping; one author used Stata for assigning students to tasks, and two others used R for producing final visualizations.

DATA AVAILABILITY

The data collected by the students and authors that support the findings of this study are available in an OSF repository:
`https://osf.io/72ajq/overview?view_only=fdc707fcf33848678a4583ae90b36514`.

SUPPLEMENTARY MATERIAL

Data details, sample instructions, and a survey template are provided in the online appendix.

References

Anderson, R. G. (1992). The GAUSS programming system: A review. *Journal of Applied Econometrics 7*(2), 215–219.

Angrist, J. D. and J.-S. Pischke (2009). *Mostly Harmless Econometrics: An empiricist's companion.* Princeton University Press.

Angrist, J. D. and J.-S. Pischke (2017). Undergraduate econometrics instruction: through our classes, darkly. *Journal of Economic Perspectives 31*(2), 125–144.

Bailey, M. (2019). *Real Econometrics: The Right Tools to Answer Important Questions* (Second ed.). New York: Oxford University Press.

Baumer, B. S., D. T. Kaplan, and N. J. Horton (2017). *Modern data science with R.* Chapman and Hall/CRC.

Bednarowska-Michaiel, Z. and E. Uprichard (2025). Bringing interdisciplinary data science education challenges into the classroom. *Journal of Statistics and Data Science Education* (forthcoming).

Best, E. and D. J. Mallinson (2024). Quantitative political science education in the past and future. *Journal of Political Science Education 20*(4), 637–649.

Carver, R., M. Everson, J. Gabrosek, N. Horton, R. Lock, M. Mocko, A. Rossman, G. H. Roswell, P. Velleman, J. Witmer, et al. (2016). Guidelines for Assessment and Instruction in Statistics Education (GAISE) College Report 2016.

Chambers, J. M. (2020). S, R, and data science. *Proceedings of the ACM on Programming Languages 4*(HOPL), 1–17.

Chonacky, N. and D. Winch (2005). Reviews of maple, mathematica, and matlab: Coming soon to a publication near you. *Computing in Science Engineering 7*(2), 9–10.

Christodoulou, D. (2015). Stata enabling state-of-the-art research in accounting, business, and finance. In E. Pinzon (Ed.), *Thirty Years with Stata: A Retrospective*, Chapter 13. College Station, Texas: Stata Press.

Conaway, B., C. Clark, J. Arias, and J. Folk (2018). Integrating econometrics: A modern undergraduate economics capstone experience. *The Journal of Economic Education 49*(3), 260–270.

De Veaux, R. D., P. F. Velleman, D. E. Bock, A. M. Vukov, A. C. Wong, and C. Burkett (2005). *Stats: data and models*. Pearson/Addison Wesley Boston.

Dogucu, M. (2024). Reproducibility in the classroom. *Annual Review of Statistics and Its Application 12*.

Ferrall, C. (1995). Interactive statistics tutorials in stata. *Journal of Statistics Education 3*(3).

Fišar, M., B. Greiner, C. Huber, E. Katok, A. I. Ozkes, and M. S. R. Collaboration (2024). Reproducibility in management science. *Management Science 70*(3), 1343–1356.

Gertler, P., S. Galiani, and M. Romero (2018). How to make replication the norm. *Nature 554*(7693), 417–419.

Hallman, J. (1993). Review of S-Plus. *Journal of Applied Econometrics 8*(2), 213–219.

Li, L., Y. Chen, M. C. Levenstein, and L. Vilhuber (2024). Crowdsourcing digital public goods: A field experiment on metadata contributions.

Munafò, M. R., B. A. Nosek, D. V. Bishop, K. S. Button, C. D. Chambers, N. Percie du Sert, U. Simonsohn, E.-J. Wagenmakers, J. J. Ware, and J. P. Ioannidis (2017). A manifesto for reproducible science. *Nature human behaviour 1*(1), 0021.

Natl. Acad. Sci. Eng. Med (2019). *Reproducibility and replicability in science*. National Academies Press.

Ostblom, J. and T. Timbers (2022). Opinionated practices for teaching reproducibility: motivation, guided instruction and practice. *Journal of Statistics and Data Science Education 30*(3), 241–250.

Perkel, J. M. et al. (2019). Julia: come for the syntax, stay for the speed. *Nature 572*(7767), 141–142.

Renfro, C. G. (2004). A compendium of existing econometric software packages. *Journal of Economic and Social Measurement 29*(1-3), 359–409.

Rodriguez, R. N. (2011). Sas. *Wiley Interdisciplinary Reviews: Computational Statistics 3*(1), 1–11.

Schissler, A. G., H. Nguyen, T. Nguyen, J. Petereit, and V. Gardeux (2014). Statistical software. *Wiley StatsRef: Statistics Reference Online*, 1–11.

Schwab-McCoy, A., C. M. Baker, and R. E. Gasper (2021). Data science in 2020: Computing, curricula, and challenges for the next 10 years. *Journal of Statistics and Data Science Education 29*(sup1), S40–S50.

Siegfried, R. M., K. G. Herbert-Berger, K. Leune, and J. P. Siegfried (2021). Trends of commonly used programming languages in CS1 and CS2 learning. In *2021 16th International Conference on Computer Science & Education (ICCSE)*, pp. 407–412. IEEE.

Tucker, M. C., S. T. Shaw, J. Y. Son, and J. W. Stigler (2023). Teaching statistics and data analysis with r. *Journal of Statistics and Data Science Education 31*(1), 18–32.

Underwood, A., A. Sichel, and E. C. Marshall (2024). Teaching reproducible methods in economics at liberal arts colleges: A survey. *Journal of Statistics and Data Science Education 32*(3), 296–302.

Van Rossum, G. and F. L. Drake (2003). *An introduction to Python*. Network Theory Ltd. Bristol.

Vilhuber, L. (2020). Reproducibility and replicability in economics. *Harvard Data Science Review 2*.

Vilhuber, L., H. H. Son, M. Welch, D. N. Wasser, and M. Darisse (2022). Teaching for large-scale reproducibility verification. *Journal of Statistics and Data Science Education 30*(3), 274–281.

Wilson, J. R. and K. A. Lorenz (2015). *Modeling binary correlated responses using SAS, SPSS and R*, Volume 9. Springer.

Wooldridge, J. M. (2016). *Introductory Econometrics: A Modern Approach 6th ed.* Cengage learning.

Wrobel, J., E. C. Hector, L. Crawford, L. D. McGowan, N. da Silva, J. Goldsmith, S. Hicks, M. Kane, Y. Lee, V. Mayrink, et al. (2024). Partnering with authors to enhance reproducibility at jasa. *Journal of the American Statistical Association*, 1–3.

Online Appendix

Not for print publication

Table A1: Data sources, sample sizes, and dates

	Sample Size	Years Gathered Min	Years Gathered Max	Figure
a. Automated Data Collection from ICPSR				
American Economic Review	1604	2010	2024	
AEJ: Applied Economics	609	2010	2024	
AEJ: Economic Policy	646	2010	2024	
AEJ: Macroeconomics	480	2010	2024	
AEJ: Microeconomics	246	2010	2024	
Journal of Economic Perspectives	233	2013	2024	
Journal of Economic Literature	60	2014	2024	
AEA Papers and Proceedings	613	2018	2024	
American Economic Review: Insights	126	2019	2024	
Subtotal	4617			3
b. Automated* Data Collection from Harvard Dataverse				
Economics Journals				
Review of Economics and Statistics	1,381	2010	2024	
Quarterly Journal of Economics	304	2016	2024	
Journal of Political Economy	142	2022	2024	
Journal of Political Economy: Microeconomics	46	2022	2024	
Journal of Political Economy: Macroeconomics	33	2022	2024	
Subtotal	1906			2
Political Science Journals				
American Political Science Review	643	2007	2024	
Political Analysis	528	2009	2024	
American Journal of Political Science	708	2012	2024	
British Journal of Political Science	537	2015	2024	
Journal of Politics	971	2015	2024	
Subtotal	3363			4
c. Manual Data Collection				
Journal of the American Statistical Association (**)	145	2017	2023	5
Journal of the American Statistical Association	218	2024		8
American Economic Review	90	2024		6
American Political Science Review	103	2024		7

Note: This table presents information on the data sources underpinning Figures 2-8. Data were gathered in two ways: systematically via the collection of metadata or file extensions; and manually by students. (*) For a small sample (68) articles in the Dataverse, we cross-validated systematic data collection with student manual data collection. (**) JASA appears twice in manual data collection to reflect the two mutually exclusive samples we draw from the journal: first a sampling of articles from 2017 to 2023, and second a comprehensive sample covering only articles published in 2024. We create these comprehensive 2024 samples for JASA, AER, and APSR, shown in panel (c). Data sourced from Harvard Dataverse, ICPSR, and journal pages.

Statistical Software Survey - Sample Instructions

The Assignment

The goal is to see how researchers working in different disciplines share materials for replication of empirical statistical analyses (and what software they use). We will visualize the results once everyone has done their small piece of data collection. For your part, this means: (a) visiting a website relating to a published paper, (b) exploring it in a structured way, (c) putting what you find into an online survey, and (d) recording in [Course software] that you have done so.

This will always begin with a "URL" or "DOI stub." If it is a URL, you can follow it directly. If it is a "DOI stub" (such as `123.456/abc/def`), you would go to the website `https://doi.org/123.456/abc/def` which will lead to an online journal or data repository. Steps after that vary, as described below.

Basic procedure for file types

The main information we will be gathering is about the extension at the ends of file-names. This extension indicates the type of the file and, typically, the kind of software with which that file is used. Occasionally there could be **other** file types or ambiguous or complex cases, and there will be a space for you to describe those. There are [N] kinds of exploration you will do:

1. (2x) **Warm-up**
2. (Nx) **Political Science/Economics**: Harvard Dataverse
3. (Nx) **Statistics**: Journal of the American Statistical Association (JASA)
4. Any other journals/sources

When you find a "replication package," "code repository," or "github page" with data and/or analysis files, there are a few main file types to look out for:

- R files (ending in `.R`, `.Rda`, and `.Rmd`)
- Stata files (ending in `.do`, `.ado`, and `.dta`)
- Python files (ending in `.py`)
- Matlab files (ending in `.m`)
- SAS files (ending in `.sas` and `.sas7bdat`)
- SPSS files (ending in `.spss` and `.sav`)

- C/C++ files (ending in `.c`)

In each case, the question is not how many of each file there are, it is simply **whether** each type of file is ever present.

If you find a statistical package **besides** those listed (e.g. FORTRAN, Julia, etc), select either the option "Other: ... " or "Ambiguous: ... " after the main file type options and describe the package in the next text field.

One tricky step with replication packages and repository entries is that they might contain **compressed** folders; these compressed folders contain more files to be investigated. Such compressed folders are often given `.zip` or `.z` or `.gz` or `.tar.gz` file extensions. In those cases, you should open the compressed folder and take note of each kind of statistical package file that you see inside it.

If a journal article is not accompanied by a replication package, or at least you do not immediately find the code, then **skim** the article to see if it mentions any of these terms which might lead you to the files: <u>software</u>, <u>scripts</u>, <u>code</u>, <u>version</u>, <u>package</u>, <u>github</u>, <u>replication</u>, <u>reproducing</u>, <u>py</u>.

Keep (rough) track of how long this takes in total

When you are done, one of the two questions in the [Course software] quiz will be how long the ten DOIs took you in total. So keep (rough) track please!

If you encounter a very time-consuming case

Occasionally a single DOI will lead to many hundreds of files, or be difficult in some other time-consuming way. Do your best, but please do **not** spend more than fifteen minutes on **any one** of the ten assigned DOIs. If it takes more than 15 minutes, stop work on that DOI, proceed by moving on to other DOIs, and describe which DOI was problematic in the [Course software] quiz when you are done.

If anything else goes wrong

If you make a mistake or find something unusually confusing about a particular entry, make a note of the DOI and/or the completion code you get at the end of the online survey, and mention either or both of them in your comments in the [Course software] quiz when you are done. This will help us resolve any issues later.

Where is the online survey?

[URL HERE]

How do I get started?

Start with the warm-ups on the next page.

A Warm-up

To familiarize yourself with the online survey, the first step is to do warm-up surveys. There are **two pretend entries** in this category, described below. All online surveys for this exercise will begin with a (pretend) "DOI stub" meaning a Digital Object Identifier. Online surveys end by producing a completion code in case you need it for referencing any problem that came up.

A.1 Warm-up 1

In the first warm-up, fill out the online survey as though you are doing a Harvard Dataverse exploration. The DOI stub you should enter into the online survey is `123.456/STUB/1/HD` (you don't really have to go there on the web, as it won't work). Imagine there you find a replication package with some R `.R` files, some Stata `.dta` and `.do` files, a `.zip` folder that in turn contains some Python `.py` files, and otherwise only text and pdf documentation.

Once you go through the survey and answer the questions accordingly, you will see a "completion code" at the end in case you need it for reference.

A.2 Warm-up 2

In the second warm-up, you should fill out the online survey as though you are exploring a JASA article. The DOI stub you should enter in the survey is `234.567/STUB/2/JASA`. Imagine you find that the article itself mentions using Python, saying ("we use Python package *autobounds* and python solver *cvxpy*"), but that even when you look at the Supplemental tab, you cannot find any replication files, just additional graphs and text. As before, once you complete the survey, you will get a completion code in case you need it.

B Harvard Dataverse

[N] of the DOI stubs assigned to you (in a list or spreadsheet provided by your professor) will be Harvard Dataverse entries. For these entries, you will be given a DOI stub. If you enter that stub (say, "`123.456/7`") into a web browser after typing `https://doi.org/` (so, altogether, typing something like `https://doi.org/123.456/7`), the browser will take you to a repository entry with replication files in it. Browse or download those files, noting what kinds of files you see, as described in the **Basic procedure for file types** above.

C JASA

[N] of the DOI stubs assigned to you (in a list or spreadsheet provided by your professor) will lead to a journal article in JASA. (Put `https://doi.org/` in front of the stub, as before.) In order to access JASA articles it is easiest to go through the campus network but if you are off-campus you should be able to access articles through the library. It may or may not have replication files associated with it. At the top of the article there are typically tabs for "Full Article," "Figures & data," "References," and "Supplemental;" it is under the "**Supplemental**" tab that one can often find a .zip file with code. If there is such a .zip file, repository, or replication package, explore it following the usual steps in the **Basic procedure for file types**.

Occasionally, a journal article will not be directly accompanied by such files. In those cases you should skim the article for key words which might lead you to either the files or a description of the files: software, scripts, code, version, package, github, replication, reproducing, py. In some cases, these keywords may lead you to discover that replication files are linked from within the article itself. If you find this link, follow it and characterize the files you find there. If you do not find this, but you find a description of the software without any link to the actual code, note what software is described, and note the text of the sentence(s) in the article indicating what software was used.

D Any other journals/sources

Any other journals or online data sources here.

E Some Examples

If you would like to see some examples that use different types of statistical software, try any of the links below.

- Dataverse entry, Stata files
  10.7910/DVN/UV8BOW

- Dataverse entry, mix of R, Stata, and Matlab files
  10.7910/DVN/HLO4XC

- JASA article, R files, linked via github mention in text
  10.1080/01621459.2023.2197686

- JASA article, mentions using R package in text
  10.1080/01621459.2020.1788949

- JASA article, Matlab files and one R file, via "Supplemental" tab
  10.1080/01621459.2022.2147074

```
Start of Block: Onboarding

 1. unix - Please provide your UNIX username

 2. doi-stub - Please provide the URL or DOI stub associated with this survey entry:

 3. any-comments - Before you answer detailed questions about this entry,
                was there anything unusual that you would like to comment on? (optional)

 4. replication-files - Were you able to find any replication files associated with this DOI?
  o Yes, I was able to identify replication files (1)
  o No, I was unable to identify replication files (2)

 Skip To: End of Block If 4. replication-files (Were you able...)
                              = Yes, I was able to identify replication files

 Display this question:
 If 4. replication-files (Were you able...)
      = No, I was unable to identify replication files

  5. read-paper - Did you find any description of the software used in the article?
   o Yes, the article described software (1)
   o No, the article did not describe software used. (2)

  Skip To: End of Survey If 5. read-paper (Did you find any description...)
                      = No, the article did not describe software used.

  Display this question:
  If 5. read-paper (Did you find any description...)
       = Yes, the article described software

   6. software - What software was used for the analysis in this article?
    o R (1)
    o Stata (2)
    o Matlab (3)
    o C/C++ (4)
    o SAS (5)
    o Python (6)
    o SPSS (7)
    o Other (8)

  Display this question:
  If 5. read-paper (Did you find any description ...)
       = Yes, the article described software

   7. article-description - What was the exact wording used by the authors in the article?

 Skip To: End of Survey If Condition: 7. article-description (What was the exact wording ...)
               Is Not Empty.

End of Block: Onboarding
```

```
Start of Block: Data Entry

 8. any-r-files - Were you able to see any R code files
    (ending in .R, .Rda, Rmd) with the replication materials?
  o Yes (1)
  o No (2)

 9. any-stata-files - Were you able to see any Stata code files
    (ending in .do, .ado, .dta) with the replication materials?
  o Yes (1)
  o No (2)

 10. any-py-files - Were you able to see any Python code files
    (ending in .py) with the replication materials?
  o Yes (1)
  o No (2)

 11. any-sas-files - Were you able to see any SAS code files
    (ending in .sas, .sas7bdat) with the replication materials?
  o Yes (1)
  o No (2)

 12. any-spss-files - Were you able to see any SPSS code files
    (ending in .spss, .sav) with the replication materials?
  o Yes (1)
  o No (2)

 13. any-c-files - Were you able to see any C/C++ code files
    (ending in .c) with the replication materials?
  o Yes (1)
  o No (2)

 14. any-matlab-files - Were you able to see any Matlab code files
    (ending in .m) with the replication materials?
  o Yes (1)
  o No (2)

End of Block: Data Entry

Start of Block: Debrief

 15. case-ambiguity - Was this case clear or ambiguous?
  o Clear: I was able to identify files and complete the task (1)
  o Ambiguous: There were aspects of this task which were complex (2)
  o Other: Unclear, but not objectively ambiguous (3)

 Display this question:
 If 15. case-ambiguity (Was this case clear or ambiguous?)
        != Clear: I was able to identify files and complete the task

  16. ambiguity-elaborate - Please briefly describe why this case was unclear:

End of Block: Debrief
```