6-3-2019

Efficient Empiricism: Streamlining Teaching, Research, and Learning in Empirical Courses

Tomas Dvorak
Union College - Schenectady, NY

Simon Halliday
Smith College, shalliday@smith.edu

Michael O'Hara
St. Lawrence University

Aaron Swoboda
Carleton College

Efficient Empiricism: Streamlining Teaching, Research and Learning in Empirical Courses

Tomas Dvorak, Simon D. Halliday, Michael O'Hara and Aaron Swoboda

The increasing importance of empirical analysis in economics highlights the need for efficient ways to bring these skills to the classroom. R Markdown is a new technology that provides a solution by integrating writing, statistical work and computation into a single document. R Markdown benefits students and instructors by streamlining teaching, research, and collaboration. We report on our use of R Markdown in undergraduate teaching, including core courses, electives, and senior theses. We discuss the costs and benefits of adoption, and explain the advantages of R Markdown in teaching reproducibility of empirical work, avoiding time-consuming and error-prone 'cut and paste,' and facilitating a one-stop solution for importing, cleaning, manipulating, visualizing and communicating with data.

Keywords: statistics teaching; econometrics teaching; reproducibility; data.
JEL Codes: A22; B4; C8

Doing economics means doing data analysis. Hamermesh (2013) demonstrates that top publishing in economics has changed over time, documenting an 'empirical turn' in economic research and publishing. Outside of academia, the shift towards using data to improve decisions has been even more pronounced (see Wegener and Sinha (2013), Brown, Chui, and Manyika (2011), BHEF (2017)). The empirical turn in economics has been matched by an increase in the offerings of colleges teaching econometrics or employing the skills of econometrics in electives (Johnson, Perry, and Petkus 2012) and matched by an increase in the demand for empirical skills by employers.[1] It is therefore incumbent on instructors to consider new ways to teach such empirical skills. Recognizing this need, Conaway et al (2018) propose a framework for an empirical capstone course for economics majors that integrates econometric methods taught in prior classes.

Teaching empirical skills, however, can come at a high cost in time and effort for students and instructors, especially in courses in which these skills are tools rather than the specific aim of a course. Fortunately, new technologies can help streamline empirical work to make teaching, research and collaboration with students easier and more efficient. We suggest the combination of R Studio and R Markdown to accomplish these goals.[2]

Economics education researchers have articulated the need for understanding data analysis, applying reproducibility protocols, and learning empirical methods in team-based learning. Building on Hansen (2001), Allgood and Bayer (2016) identify essential competencies of students graduating with an economics major. Of particular interest to us is the first essential competency that, "[s]tudents should know how to ask an economic question, gather information and resources, form an explanatory hypothesis, collect data that can be used to test the hypothesis, analyze the data, draw conclusions, and suggest future research" (Allgood and Bayer 2016, 106).

To engage fully with the competency proposed by Allgood and Bayer requires that students make all of their work fully reproducible. Reproducible student work benefits both students and instructors. When students make their work reproducible, they learn how to organize their work, how to make the results of the work transparent, and how to make it easier for others to interpret their work. Students therefore learn hands-on skills that are valuable outside of the economics classroom as literate programming and reproducible research make students more effective and employable after graduation. In a broader view, it gets students to engage in the methods of economics as a social science in which results are valid only to the extent that they are reproducible. Reproducible work also benefits instructors, as it is much easier to find and correct errors and guide students in their research when everything they have done is documented and transparent.

This paper proposes R Markdown as the best tool for reducing the barriers to entry-level coding for undergraduate students, thereby enabling reproducible empirical work and marking an important step towards attaining the first essential competency articulated by Allgood and Bayer (2016). R Markdown allows the integration of data processing, analysis, table and figure creation, and document write-up in a single document and thereby avoids the time-consuming and error-prone copy-and-paste workflow inherent in most existing methods (Baumer et al 2014). For the instructor, R Markdown facilitates the production of class materials and eases grading. In an empirical course, instructors often need to produce many documents: empirical examples to use in class, slides for lecture, class or lab exercises, and homework assignments and solutions. R Markdown documents easily create all of these formats, reducing the time cost of producing different materials. In addition, the use of R Markdown makes all of this work inherently reproducible. In fact, it is difficult to do work in R Markdown that is *not* reproducible. R Markdown also "plays nicely" with tools like github, R Studio server, and R Studio Cloud, which make it well calibrated for teaching and collaboration. For example, in R Studio Cloud and R Studio Server, an instructor can share projects with students or students can share projects among themselves, which enables collaboration on projects.[3] As a final benefit, R Markdown and R Studio are both open source and free, vastly expanding access for students when in school and beyond.[4]

We illustrate the use of R Markdown in a variety of courses at four different campuses to show that R Markdown is an efficient way of teaching reproducible empirical methods at the undergraduate level. Drawing from our own experiences, we share our strategies for incorporating R Markdown and reproducibility into courses that range from core econometrics courses, to electives, to senior seminars. We share student reactions and how to avoid certain pitfalls when introducing R Markdown. We also address common criticisms of switching from other packages, such as Stata and SPSS, to R Markdown and argue that using R Markdown remains a robust alternative.

LITERATURE: REPLICABILITY, REPRODUCIBILITY AND EMPIRICAL PEDAGOGY

Our focus in this paper is on *reproducibility,* which requires that the original results of a study can be repeated, as distinct from the issue of *replicability*, which involves verifying the robustness of a result to changes in data or methodological decisions. Important research findings in economics have often failed both basic reproduction and replication, the most notorious of which has become the "spreadsheet error" underpinning Reinhart and Rogoff (2010). In the process of trying to reproduce the original results, Herndon, Ash, and Pollin (2014) discovered data errors and omissions, that, once corrected, failed to produce the original results.

In recent years, reproducibility and replicability have become significant concerns in economic research (Camerer et al. 2016; Christensen and Miguel 2016; Berry et al. 2017; Chang and Li 2015) and affiliated social sciences (Open Science Collaboration 2015). Recent work has attempted to provide tools for diagnosing what a specific reproduction or replication achieves or fails to achieve and whether it invalidates or supports the original paper (Clemens 2017): for example, we need to be able to discuss whether and how a replication finds the same sign for a coefficient as previous research, but a different magnitude or level of statistical significance; with respect to reproducibility, the availability of original data and code, and sufficient document enables other researchers to reproduce the original work and subsequently calibrate a potential replication properly. These concerns are part of a broader movement toward greater transparency and integrity in the social sciences, as embodied in organizations such as the Berkeley Initiative for Transparency in the Social Science (BITSS), and the Association for Integrity and Responsible Leadership in Economics and Associated Professions (AIRLEAP), among others.

To this end, Ball and Medeiros (2012) introduce the TIER (Teaching Integrity in Empirical Research) protocol as a way to foster reproducibility in undergraduate data work. The TIER protocol prescribes a step-by-step set of instructions for the student to follow in order to ensure full reproducibility of results. The process assumes students will pass data through multiple programs like Excel and Stata repeatedly, requiring the student to write and organize multiple scripts and intermediate data files.

Knittel and Metaxoglou (2018) similarly present a project organizational structure and set of coding best practices for Stata and LaTeX analysis and write-up. They also describe efficient ways to pass information among programs (for example, Stata for analysis, then Excel for table creation, and finally LaTeX for inclusion in a write-up). The complexity of this system and need for mastery of multiple tools makes such a system more appropriate for professional economists already familiar with LaTeX and/or advanced undergraduates working as research assistants for whom there is sufficient time for a positive return on the initial investment in start-up costs.

A number of papers have examined how to introduce empirical, econometric and analytical skills in senior electives, capstone courses, and team-based learning courses, among others (Conaway et al 2018; McGoldrick 2008; Klein 2013; Imazeki 2015), however their emphasis has not been on reproducibility. Adding reproducibility to an already full curriculum can create a heavy burden on both instructors and students. Our paper contributes to this literature by focusing on how to ease that burden. We provide tools and methods to teach reproducible empirical research by increasing the efficiency of instruction, particularly at the undergraduate level. We have provided a numbered online appendix at simondhalliday.com/efficient_empiricism to provide supplementary information.

HOW DOES R MARKDOWN WORK?

R Markdown and R Studio make up an integrated environment that combines writing, statistical work and computation in one document.[5] An R Markdown document is a combination of ordinary text and statistical code. The text is what we normally call "the paper" (or "prose" in R markdown jargon). The statistical code comprise pieces of R code (in R Markdown parlance they are called "code chunks") that run the analysis. Figure 1 shows an example of a source R Markdown file with the output of that source code produced in Figure 2.

Figure. 1 About here:

Figure 2 About Here:

The example shows a few lines of R code surrounded by plain text. The plain text can include tags for formatting. In the example, hashtags (#s) are tags for section and subsection headings: one hashtag for each level of heading. Tags for italics, lists, equations etc. are also available. While the text is not "what you see is what you get", it is highly readable. In our experience, students find using R Markdown far easier than other markup languages such as LaTeX or html.[6] R Markdown can include tags for references linked to a bibliography source file. Our students have used these extensively and appreciate the automatic generation of a reference list in the correct format required for grading.[7] Furthermore, using the **rticles** package or other template, the student's paper can be formatted to look like a published paper without the student having to put any work into the formatting.

The "code chunks" in R Markdown can contain any R code. It can be code that retrieves, merges or appends data; calculates descriptive statistics; estimates regressions; or, as in the above example, creates a figure. The code can be executed chunk by chunk, which is what normally happens as researchers build and test their code, or it can be executed all together from start to finish.

The text and code are put together by "knitting" the R Markdown source document. Knitting the source document executes all of the R code and formats the surrounding text. Knitting produces an output file in which the text is properly formatted, and which includes any tables and figures generated by the R code. R Markdown can produce the output file in a variety of formats including html, pdf and MS Word (.docx). Figure 2 shows the knitted version of the R Markdown shown in Figure 1 and it demonstrates how the plain text has now been formatted and that a figure was produced.

A related virtue of R Markdown for teaching is that R Studio interprets R Markdown documents in the manner of a laboratory notebook: when you write a code chunk and execute that code chunk within the R Markdown document, R Studio produces a preview of the output within the R Markdown document. Students therefore see what the knitted statistical and graphical output will look like prior to knitting the document to see the output of all of their work.

HOW IS R MARKDOWN EFFICIENT?

The integrated environment provided by R Markdown and R Studio has several advantages over traditional systems of package suites. We address some of these with a focus on the benefits to both students and instructors.

One-stop solution

R Markdown integrates analysis with text, tables and figures. The entire paper is produced (including tables and figures) when the source document is knitted. This contrasts with the traditional way of doing empirical analysis which involves switching between statistical (Stata, SPSS, R), word-processing (MS Word, LaTeX, Google Docs), and perhaps presentation (Powerpoint, Beamer) software. In a more traditional setup, analysis is done in the statistical software and its output is copied and pasted into the word-processing or presentation software. This is time-consuming and prone to errors. For example, a minor change in the data set can set off a cascade of minor changes in all of the tables and figures, which then have to be re-copied and re-pasted.

From a pedagogical perspective the integration of text, tables and figures, makes it more likely that students will experiment with and explore their data more deeply. When tables and figures are automatically reproduced, students are more likely to see how their results differ, for example, if they drop small countries from a sample. This fosters curiosity and learning about the subject they are studying.

From the instructor's perspective, the integration of R Markdown greatly streamlines the creation of materials for teaching empirical topics. Code that is written in Markdown to show an example in class can simply be knitted into slides or documents to provide to students as notes or posted as a web page. An exercise used in class can be slightly changed to import different data and knitted as a homework assignment. The instructor saves a great deal of time spent producing different materials with related content.

Grading is also easier since exercises can have the instructions and questions in the same document as the students' answers and the code they used to compute the answers. This makes it much easier to locate the source of errors and find areas of misunderstanding than in the case where an instructor reads a written assignment and has to go back to a separate code file to try to find why mistakes appear. And if reproducibility is a component of the grade, this becomes almost a non-issue.

R Markdown is reproducible

R Markdown makes data analysis inherently reproducible because every step in the analysis is documented using code that is part of the source R Markdown document, and therefore part of the "paper." There is no other program that manipulates or analyzes the data – everything is done within R Markdown. Therefore, if provided with the data, a third party who is unfamiliar with the project could use the source R Markdown document to see how the analysis was conducted and independently reproduce the work.

Our students take it as a given that, for example, if they decide to drop small countries from their cross-country sample, it must be documented. R Markdown makes this a non-issue. The code to drop those countries must be part of the document, otherwise, the tables and figures would be incorrect. In fact, students who are first introduced to empirical work through R Markdown usually can't imagine doing it any other way. If instructors wish to highlight the problem of reproducibility in traditional methods, they can assign students to attempt to reproduce results from papers that are not documented in this way, as one of the authors discusses in a case study below.

We emphasize to our students that the data their R code uses must be in its original form, i.e. not having been manipulated previously by hand or by other programs. Fortunately, it is becoming common practice to retrieve data from standard databases using code. For example, the World Development Indicators can now be retrieved using the **wdi** package, FRED and other financial data using **quandl**, Census data using **tidycensus** and **acs**, and IPUMS data (including CPS) can be directly retrieved using **ipumsr**.

For the instructor, the assessment of reproducibility also becomes almost automatic. The authors take as given that all student work must be fully reproducible. To enforce this, student work must be reproduced. The authors have required reproducibility of student projects using other packages in the past and it can be very time consuming to piece together the code files and get them to run to verify the results in the paper. R Markdown makes reproducing student work relatively trivial. Students turn in their R Markdown file. The instructor opens it and runs it. If the students used this code to produce their paper, there is little reason that it should produce anything different when the instructor runs it.

R Studio is well calibrated for collaboration

The combination of R Studio and R Markdown is well calibrated for collaboration between the instructor and students, and among students themselves in research teams. R Markdown and R Studio improve teaching and maximize the efficiency of the instructor's time by lowering costs to the instructor via effective debugging and error detection tools and by easing collaboration through R Studio Server and, more recently, R Studio Cloud.

R Studio has very good syntax error-detection capabilities, letting the student know *that* an error exists (existence), *where* an error originates (origin), and *what* likely kind of error it is (diagnosis). The authors have all experienced cases where a student has forgotten an opening or closing parenthesis or quotation mark, misplaced a comma, or made a similar error of syntax. When instructors become heavily involved in detection and diagnosis of errors, valuable one-on-one instructor time gets consumed.

R Studio also provides an R Studio Server edition (free to educational institutions) that students can access with their university log-in credentials through a web browser, and more recently a cloud version as well. In R Studio Cloud and R Studio Server, an instructor can share projects with students or students can share projects among themselves, which enables easy collaboration on projects. R Studio can also be integrated with github to enable collaboration and version control. We have also found that using R Studio Server expedites initial take-up with students later installing R and R Studio on their own devices once they are comfortable with the software.

R Studio, R and R Markdown are free

Lastly, R, R Markdown and R Studio are all free. Students must pay for tuition, textbooks, related educational materials and software. Finding ways of alleviating constraints on them has become a priority both for faculty and for college administrations.[8] The free and open source model also makes R and R Markdown increasingly the software of choice for many businesses,[9] making our students more marketable after graduation.[10]

WHAT ARE THE COSTS AND ALTERNATIVES?

Whenever an instructor wishes to adopt a new computer language or technology there are costs involved in learning and adapting existing courses to use the new technology that must be weighed against the benefits. While we believe the benefits to efficiency discussed above far outweigh the costs, these costs still need to be addressed. Here we delineate some of the common objections we have encountered. We also discuss some existing alternatives and explain why we believe the combination of R Markdown and R Studio are superior to other currently available options.

Learning to code

Perhaps the key hurdle in teaching reproducible empirical methods is the need for some degree of coding skills. Writing code is a skill set of its own, and within the context of an economics course, it is a means to an end rather than the main focus. R Markdown allows students to be introduced to coding gradually, and to be fully integrated into the content of the course, rather than taught as a separate topic. At the beginning of the course, students can be provided with pre-written R Markdown that contain the code they need to perform tasks, we provide examples of such templates in Online Appendix H.

The integrated nature of the Markdown format allows the code to be combined with comments about what each line does as well as longer explanations in the prose outside of the code chunks that explain the purpose of the code in the overall analysis. At this stage, students may only be expected to run the code and see the results. Instructors can then proceed to having the students change small parts of the code, such as adding new variables to a regression model or making changes to a dataset. Step by step, the students become familiar with the code, enough to be able to write code chunks to perform the basic tasks required by an analysis.

In Table 1 we list the competencies students may need when coding empirical analyses. The elementary set of competencies includes loading the data, producing descriptive statistics, and estimating a regression. Each of these competencies can be achieved by one line of code in most statistical packages. R Markdown integrates the code and the output of the code into the narrative of the analysis. Just as students are accustomed to edit their prose, they naturally become accustomed to edit their code, and see how results change. The ability to adapt, reuse and combine code allows them to do increasingly sophisticated analyses.

Table 1 about here

Table 1: Student coding competencies

In programs such as Stata, analysis can be conducted with drop down menus which then triggers a display of the code that would perform the same task. This code can then be copied and pasted into a script file (a Stata .do file). While this provides significant aid in learning code syntax, it is not clear whether learning a sequence of dropdown manipulations is any less burdensome than learning the code itself.

Furthermore, to make the work as whole reproducible requires that the Stata .do file and any prose that analyzes the data be wrapped together by a batch file that runs the file producing the output and the file producing the prose, e.g. a LaTeX file. Such a process requires instructing students on batch scripting in addition to teaching them about the language itself (this process is advocated by Gentzkow and

Shapiro (2014) for professionals and advanced research assistants, but we think it is infeasible for medium to large undergraduate classes).

R Markdown and R Studio reduce the cost of teaching and learning coding skills which are necessary for empirical coursework in economics and beneficial for the job market. Specifically, a student uses templates for initial code, can immediately see the output she creates, and works incrementally on existing code she knows works to obtain new output. This feature is critical since student's experience writing computer code varies substantially, with many students having no prior experience.

The learning curve

Until recently, R lacked an easy to use, dedicated integrated development environment (IDE). The IDE provides access to a command line window, a script or R markdown window (with tabs for different files the user is working on), the environment window with tabs for history and github access, and the context window with tabs for files/folders, packages, plots produced by commands, help for commands, and the viewer for interactive output. With the release of R Studio in 2011, R became much more accessible providing an intuitive and user-friendly way for users to write and execute code, browse data, manage files, search help, view plots, do version management, and more, all inside one application.[11]

In addition to R Studio, the transition to R has recently been made much easier through the introduction of many useful and user-friendly packages such as **tidyverse**, **mosaic**, and **stargazer**. The first two provide convenient functions for data manipulation, exploration, and visualization, while the latter creates automatically formatted regression tables. We provide additional suggestions about the recommended packages for teaching R and R Markdown in Online Appendix C, highlighting the value of the tidyverse, stargazer and mosaic packages (with student capabilities as described in Table 1 and program capabilities as in Online Appendix J). The **remedy** package also makes teaching the syntax of R Markdown easier as it provides menus to show the syntax required for particular formatting, therefore easing student learning.

Stata and Dyndoc

Most of the authors are Stata users and gleefully anticipated the release of Stata's markdown options that were included with the Stata 15 release. We have investigated the features available in Stata and we were able to produce viable documents. However, we conclude that Stata's markdown falls short of its R Markdown alternative for a number of reasons: First, Stata's dynamic document, **dyndoc**, is not integrated into Stata's development environment. For example, Stata's do-file editor is not built to handle dyndoc documents. The do-file editor treats everything as Stata code. This is good for the code part of the document (the code-chunks), but it is not good for the prose. The do-file editor does not have spell check, does not even recognize the tags, i.e. it does not help with closing of the tags (which R Studio automatically can do with parentheses, square brackets, braces, and more).

Second, some of the **dyndoc** tags are clumsy. For example, the start and end of a code-chunk use two lines each; including graphs requires one code-chunk to create a graph, and another code-chunk to include it into the document.

Third, unlike R Markdown, **dyndoc** does not produce a pdf. Instead, a separate command, **putpdf**, which uses completely different markup language is required to produce a pdf document. This is in sharp contrast to the ability of R Markdown to create many output types.

Fourth, **dyndoc** does not support LaTeX, BibTeX or even **outreg**. Lacking these tools will be a deal-breaker for many users. Fifth, **dyndoc** does not support publishing html to the web. Stata's dyndoc lacks the one-button experience of publishing html that exists in R Studio. Moreover, the html created by **dyndoc** is not self-contained (it needs any image/figure files). This makes sharing the document cumbersome, unlike the self-contained html created by R Markdown. Finally, **dyndoc** does not support code-folding, tables of contents or themes. This is mostly aesthetic, and one can certainly customize the html style sheet, but in contrast to R Markdown makes creating slick-looking documents very easy.

We freely admit that some capabilities that exist, for example, in Stata, are not as readily accessible in R and R Markdown. A common objection that Stata-users have made in the past has concerned, for example, the easy specification of robust standard errors using ", robust" at the end of a regression specification. Though this was once more challenging in R and required that the user implement a variety of other packages, now user-written packages (for example, **estimatr**) emerge to deal with these issues because R is open source and customizable. A user will often find a command or function that they want to use is not as freely available in R, but that it may be under development or recently available as a new package.

Class size

The classes in the set of case studies discussed below range from 10 to 35. But R Markdown and R are taught in a variety of settings with much larger class sizes. For example, at Duke University in the introductory statistics classes, which all use R and R Markdown, the class size ranges from 80 to 120 students. Class sizes are similar at other colleges and universities that teach with R and R Markdown, for example, Smith College has an introductory statistics class that teaches between 75 and 85 students and UCLA has similarly large class sizes. Furthermore, at the institution at which one of the authors teaches, Python (a popular introductory programming language) is taught in a class of 90-120 students in introductory computer science. Though we acknowledge the benefits of smaller class sizes, and especially small labs, for teaching R Markdown and R, we believe that the technology can be scaled. Admittedly, if a class has several hundred students, then teaching reproducible coding may be infeasible, which would be true for R, Stata, or SPSS.

CASE STUDIES

Each of the case studies below was undertaken at a liberal arts college (LAC). The class sizes vary and we comment on them for each case. We specify the role of R and R Markdown and the expected entry level of competency for each course. Each of the courses has been influenced to some degree by the concerns of reproducibility in social science research and the instructors have used the TIER protocol (or some variation thereof). We indicate whether a textbook was used for the course's main materials and if additional supplements were used. The syllabi of the courses are available in Online Appendix A. As R and R Studio are under continual development, we also comment on the resources we have used to assist in student learning and provide additional resource in Online Appendices.

Econometrics

The class is an upper-level elective. The pre-requisites for the class are an entry-level statistics class, as well as a 200-level course in Quantitative Methods (QM), where students are introduced to basic regression work using Microsoft Excel. Therefore, all students have seen regressions and basic statistics, although their competency in these skills spans a wide range. Generally, about half of students have had at least some experience with R, primarily through a statistics class in applied regression. Very few have any previous experience with R Markdown. Class size is limited to 16 in order to make use of an available computer lab with that number of machines (class size ended at 15). With some flexibility, the format of the course alternates one class period of lecture with one class period of applied work in the computer lab.

In lecture classes, the instructor uses the R Notebook format of R Markdown, so that explanations of the concepts can be combined with code to illustrate them. The Notebook format is very useful for class as one chunk of code at a time can be run and the results seen immediately. The Markdown code is written so that it can be knitted into slides using the **slidy** package. This allows students to follow along with the exercise in class and see the code being used, and then knit the document and have the notes available for study afterward.

For the applied lab assignments, students hand in hard copies of their R Notebook file in html or Word format rather than the R Markdown (.Rmd) file itself. The code is all shown in blocks, with the questions and student answers in text. The Notebook format facilitates grading, as the structure of the exercises is standardized and the code is visible, making it easy to see where misunderstandings arise. With a little strategy, instructors can format the Markdown file to make the assignments easy to read. For example, the Markdown file can be formatted with specific places for student answers and these can be formatted into boldface to make them easier to find.

The semester culminates in student projects completed in groups of two to three students. These projects are required to be conducted in R Markdown and to be fully reproducible from raw data to final results. To introduce students to this idea, their first assignment is to reproduce the results from a paper written by students the previous semester. The instructor chooses some of the best papers from the previous semester as a way of setting expectations high. Out of two semesters and 19 total projects, there was only one project in the first semester that the instructor was not able to reproduce, and this was due to not being able to get the right files from the students after they had left for winter break.

Behavioral Economics

The class is an upper-level elective with between 20 and 35 students. The pre-requisites for the class are intermediate microeconomics and a combined one semester introduction to statistics and econometrics that concludes with multiple regression. The instructor expects that students will be beginners with R, but some students enter with proficiency. Exercises in R and R Markdown were paired with relevant economic data to consider questions in economics and behavioral economics. Students confronted different data in lab exercises (experimental data, household data, macro data), giving them the opportunity to think about the kind of data they should like to analyze in their team research projects.

Having used R and R Markdown in these contexts, students were equipped to reproduce the results from published papers. After being assigned to teams, students decided on papers they wanted to reproduce and for which they would provide a new experimental design.[12] In most cases, students were able to *reproduce* fully the results of the paper, in other situations the instructor had to re-tailor their research either because they had taken on a project that was in fact too hard to reproduce in the time available or because some portion of the data was unavailable to the students. In Online Appendix H we provide a full list of all papers that could be reproduced along with some comments about the full or partial reproducibility of each paper. In all cases, R Markdown enabled student engagement with reproducibility through an easy workflow, being able to work on the project in their own time on their own devices, and through the one-stop-shop nature of R Markdown.

Having reproduced (partially or fully) the published results, students better understand how important reproducible research is mainly as a consequence of struggling to reproduce published research with published reproduction documents. They are also motivated to think about how they could improve existing research (were funds and time available to do so) and provide a replication study along with addressing a new research question while also explaining how they would ensure reproducibility of their replication. The opportunity to improve existing research and to create new research questions was the final important pillar of the students' team projects: what would they do if they could implement their own survey, run their own experiment, or design their own study? How would they test their data? Many students come up with fascinating alternatives to existing studies and demonstrate their ability to think about the research process and about their participation in a community of researchers promoting transparent and reproducible research.

Advanced Topics in Housing Economics

The class is a senior-level elective seminar capped at 15 students. Students are required to have completed intermediate "core" courses in the economics major in microeconomics, macroeconomics, and econometrics.

Past iterations of the course were primarily aimed at helping students develop their own research proposal, but experience showed that the students were not well prepared for their independent research project. Students had never collected, managed, and analyzed large datasets. They tended to use MS Excel to clean and manage their data, and moved between several software packages for data cleaning, analysis, write-up and bibliographic management. This led to mistakes (some known and others likely unknown), and students were frustrated by suggestions of exploring the sensitivity of their results when doing so required so many steps. The most recent iteration sought to help students to prepare better for the independent research project and to expose them to the important academic and ethical debates surrounding reproducible research through the use of a reproduction project.

The reproduction project is based on the primary results of Harrison and Rubinfeld (1978) and the subsequent articles O. Gilley and Pace (1996) and Pace and Gilley (1997). These papers perform a hedonic regression to explore the effect of air pollution on housing prices in Boston in the 1970s. Students were asked to reproduce some of the more accessible results of the papers following the TIER protocol (Ball and Medeiros 2012). That is, they had to get the computer to produce the correct numbers *and* show how they made the computer produce those numbers.

Students reacted in diverse ways to the learning experience. Some students voluntarily implemented the TIER protocol and used similar tools and techniques in their subsequent independent research project. Others were happy to jettison the new tools and return to their old habits. However, more than one student regretted abandoning the R Markdown suite when, late in their own research project, they encountered problems similar to those posed in the labs the previous term, and found themselves facing the more difficult task of correcting the problem without the code to enable reproducibility or the efficiency in package suite or workflow that R Markdown offers. One student told me afterwards at the senior research poster presentation, "You tried to warn me, but I didn't listen. I would advise future students to learn from my mistake!" Another commented, "It was pretty easy to take the work from my paper and make the poster because I had all of the code necessary to produce everything!"

Business Analytics

The class is an upper-level elective with between 25 and 35 students – mostly economics majors. The pre-requisites include either an introductory econometrics course or a statistics course that covers regression analysis.

The course has three learning objectives: ability to manipulate data (downloading, merging and reshaping data); ability to analyze data (this includes exploratory data analysis, visualization, as well as machine learning algorithms such as regression, nearest neighbor, naive Bayes and decision trees); and the ability to formulate questions that can be answered using data (for example, improve marketing, pricing, investing capital, customer satisfaction, fraud detection etc.). We pay special attention to validating predictions using the train and test regimen common in data science and machine learning fields.

Working with R Markdown is a key component of the class. In fact, each class topic is organized around one R Markdown document. Students bring laptops to class, and open the relevant R Markdown document. The document includes text that explains what we are doing and the code that implements what needs to be done, so students do not need to switch back and forth among applications. Students execute the code along with the instructor, and see the results on their own screens. The R Markdown document includes many in-class exercises where the instructor pauses and asks students to write their own code – usually an application of the material just covered. This is a highly engaging way for students to interact with the material. Assessment in the course is also done via R Markdown. Both problem sets and exams are done as R Markdown documents. Students are sent R Markdown documents with questions and usually a link to data. Students answer questions by doing the relevant analysis, writing code and interpreting results, all in one R Markdown document, demonstrating the efficiency in workflow and instruction of R Markdown. Students are generally very comfortable typing exams on their own laptops. They are allowed to use any notes, even search the internet, therefore simulating a real-world environment.

The learning outcomes on the part of students are very positive—both in terms of course evaluations and student performance in subsequent courses. For example, students who take this course seem to write more sophisticated senior theses, largely because the y have the ability to manipulate a greater variety of data, and combine this data in interesting ways.

Thesis, Seminars and Special Studies advising

The authors have used R Markdown and reproducible research protocols in senior theses, seminars and semester-long special studies.

The students at one college are all required to write a senior thesis. One of the authors led a seminar in Environmental and Resource Economics that consisted of nine students.[13] The college has an open curriculum with few strict prerequisites. Therefore, the students varied greatly in their backgrounds with only one student in the group having taken econometrics. Most students had little prior computer experience. There was no prerequisite of having taken an environmental economics course.

Because of the inequality in backgrounds among the students, class time was mixed between reading and analysis of papers in environmental economics and learning research methods, including data wrangling and tidying, exploratory analysis and visualization, and, as needed, some basic econometrics.

The variety in student backgrounds resulted in a wide variety of papers. The primary requirement was that the papers be empirical in the sense that they had to use data to support an argument, though this did not have to take the form of formal econometric analysis. Some students made arguments using creative data visualization methods. The **ggplot2** package (part of the **tidyverse**) provides impressive tools for producing informative and attractive visual representations of data. A few students used regression techniques to refine the visual analysis. All nine papers were fully reproducible, an impressive performance starting from students having very little computer or data experience coming into the course. The documentation for all projects is available online and available in Online Appendix C: Examples of Reproducible Research Projects.

One of the instructors has used R and R Markdown to supervise special studies. The special studies involved students first reproducing the work in published papers, then applying the methods they learned to new data. One student used Carr and Mellizo (2013) as the basis for her special studies where she worked with the 2004 Workplace Employment Relations Survey (WERS), then extended the analysis to some initial analysis of the 2011 wave of the survey (with slightly different questions). A second student was interested in the Alaska Permanent Fund and she worked on extending the work of Hsieh (2003) with new Consumer and Expenditure Survey data, though her analysis could not reproduce the complexity of the original author's work (the full work was beyond the scope of an undergraduate special studies).

Student performance

The key outcome of teaching courses with R Markdown is that student work is fully reproducible – all data manipulations are well documented by default. This is very different from our experience teaching similar courses without R Markdown. For example, in introductory econometrics we took student regression results as a given – never knowing for sure what manipulations lead to these results or whether the coefficients the students reported were the actual coefficients.

Although we don't have any evidence of causal effects (as students self-select to our courses), observationally we tend to see stronger performance in subsequent courses. This is true particularly for senior theses, as students who are familiar with coding and R Markdown tend to write more sophisticated theses as they have the ability to combine and manipulate data in new and interesting ways. Also, informal comments we have received from students about their job market outcomes indicate that knowing R helped them differentiate themselves from other economics majors. Student comments collected in Online Appendix I also suggest that students appreciated learning R and valued the opportunity to use it for labs, research and data visualization.

CONCLUSION

The increasing importance of empirical work to economics requires that we carefully assess the tools available to make this work as efficient as possible. In arguing for the use of R Markdown and R Studio, we highlighted the significant gains in terms of workflow, reproducibility, teaching, collaboration and monetary cost.

Through understanding the useful aspects of R, R Studio and R Markdown, we can think more clearly about the ways in which we practice data analysis in economics in a transparent and reproducible manner. In being so equipped, students can undertake high quality data visualization and analysis. Getting students to engage in reproducible research also permits students to practice their new skills, to become a genuine part of a research community through testing others' work, and to unleash their creativity through devising new ways to think about their own research questions for the future.

NOTES

[1] Biddle and Hamermesh (2016) nonetheless show that good theorists obtain a wage premium over non-theorists.

[2] R Studio is the integrated development environment – the program showing the command line, scripting files, folder structure, tables and plots – that most R professionals use to interact with R and R Markdown. R Markdown is a file format one can use in R Studio that acts as a lab notebook, working file for a term paper, thesis or team project, and more.

[3] A more sophisticated alternative that is taught at Duke University by Mine Çetinkaya-Rundel in a first statistics class and which is used by some of the authors with research assistants is to use github with R Studio to enable collaboration and version control.

[4] One alternative to R and R Markdown would be the potential of using Jupyter Notebooks with Python. Though this pairing satisfies the concerns we have about accessibility, they are not taught or geared towards the instruction of econometrics and statistics, whereas R and R Markdown are strongly geared towards quick adoption and to education of statistics, data visualization and econometrics.

[5] R Markdown and R Studio are separate pieces of software but we don't know anyone who uses R Markdown but not R Studio. The environment can be a desktop application or online via an R Studio Server which allows students to do their work via an internet browser through a server hosted on campus or with R Studio Cloud, an open R Studio server hosted by the R Studio company online (currently in the alpha stage of development).

[6] R Markdown follows standard markdown syntax as defined by Gruber (2012) which was designed with readability in mind.

[7] Students can copy and paste the bibliographic information provided by Google Scholar into their bibliography file and use the labels provided by Google or define their own labels.

[8] For example, one of the college's in the case studies has an ongoing commitment to encouraging the use of open educational resources to alleviate costs to students.

[9] The 2015 Microsoft acquisition of Revolution Analytics (an R implementation firm), and 2018 IBM acquisition of Red Hat (a Linux implementation firm) are examples of the value businesses put on open source products. More generally, the 2017Burning Glass Technologies report (https://www.burning-glass.com/wp-content/uploads/The_Quant_Crunch.pdf.) which highlights the need for data science skills particularly R, Python, SQL, SAS. A systematic tracking of popularity of various statistical software using job postings on indeed.com, mentions in scholarly articles also shows steep rise in R and Python at the expense of SAS and SPSS (see http://r4stats.com/articles/popularity). For more examples of businesses using R see https://www.northeastern.edu/levelblog/2017/05/31/big-companies-using-r-data-analysis/. http://r4stats.com/articles/popularity/

[10] For example, after taking the behavioral economics course using R and R Markdown, one student was hired as a paid intern, then asked to take a leave of absence from her studies to work for the company for a year in their research and development department. This is but one of many instances in which similar events occurred for instructors of these courses.

[11] To be sure, there are now a variety of other IDEs for R, but we believe that R Studio is the best IDE for beginners. One strong competitor is Jupyter notebooks that can be used with R as well as other programming languages, such as Python, Julia and Scala. R Studio can also be paired with Python and Stata within code chunks. Each offers a variety of benefits, but we recommend R Studio for beginners.

[12] Students did not engage in replication because, as a result of the class being a behavioral economics class, students were looking at laboratory data and were unable (due to time constraints) to create their own data.

[13] Two of the authors do research in and teach environmental economics, so this research seminar should not be confused with the class described earlier.

REFERENCES

Allgood, Sam, and Amanda Bayer. 2016. "Measuring College Learning in Economics." *Improving Quality in American Higher Education: Learning Outcomes and Assessments for the 21st Century*. John Wiley & Sons, 87–134.

Ball, Richard, and Norm Medeiros. 2012. "Teaching Integrity in Empirical Research: A Protocol for Documenting Data Management and Analysis." *The Journal of Economic Education* 43 (2): 182–89.

Baumer, Ben, Mine Çetinkaya-Rundel, Andrew Bray, Linda Loi, and Nicholas J. Horton. 2014. "R Markdown: Integrating a Reproducible Analysis Tool into Introductory Statistics." *Technology Innovations in Statistics Education* 8 (1).

BHEF. 2017. "Investing in America's Data Science and Analytics Talent: Call for Action." Business Higher Education Forum. http://www.bhef.com/sites/default/files/bhef_2017_investing_in_dsa.pdf.

Biddle, Jeff E., and Daniel S. Hamermesh. 2016. "Theory and Measurement: Emergence, Consolidation and Erosion of a Consensus." Working Paper 22253. Working Paper Series. National Bureau of Economic Research.

Brown, Brad, Michael Chui, and James Manyika. 2011. "Are You Ready for the Era of 'Big Data'." *McKinsey Quarterly* 4 (1): 24–35.

Camerer, Colin F., Anna Dreber, Eskil Forsell, Teck-Hua Ho, Jürgen Huber, Magnus Johannesson, Michael Kirchler, et al. 2016. "Evaluating Replicability of Laboratory Experiments in Economics." *Science* 351 (6280).: 1433–6.

Chang, Andrew C, and Phillip Li. 2015. "Is Economics Research Replicable? Sixty Published Papers from Thirteen Journals Say 'Usually Not'."

Christensen, Garret S., and Edward Miguel. 2016. "Transparency, Reproducibility, and the Credibility of Economics Research." Working Paper 22989. Working Paper Series. National Bureau of Economic Research.

Clemens, Michael A. 2017. "The Meaning of Failed Replications: A Review and Proposal." *Journal of Economic Surveys* 31 (1): 326–42.

Conaway, Brooke, Christopher Clark, J. J. Arias and Jessie Folk. 2018. "Integrating Econometrics: A Modern Undergraduate Economics Capstone Experience." *The Journal of Economic Education*, 49 (3): 260-270.

Gentzkow, Matthew, and Jesse M Shapiro. 2014. "Code and Data for the Social Sciences: A Practitioner's Guide."

Gilley, Otis, and R. Kelley Pace. 1996. "On the Harrison and Rubinfeld Data." *Journal of Environmental Economics and Management* 31 (3): 403–5.

Gruber, John. 2012. "Markdown: Syntax." *URL http://Daringfireball. Net/Projects/Markdown/Syntax.* Retrieved on 24 June 2018.

Hamermesh, Daniel S. 2013. "Six Decades of Top Economics Publishing: Who and How?" *Journal of Economic Literature* 51 (1): 162–72.

Hansen, W Lee. 2001. "Expected Proficiencies for Undergraduate Economics Majors." *The Journal of Economic Education* 32 (3). Taylor & Francis: 231–42.

Harrison, D, and D Rubinfeld. 1978. "Hedonic housing prices and the demand for clean air." *Journal of Environmental Economics and Management* 5 (1): 81–102.

Herndon, Thomas, Michael Ash, and Robert Pollin. 2014. "Does High Public Debt Consistently Stifle Economic Growth? A Critique of Reinhart and Rogoff." *Cambridge Journal of Economics* 38 (2). CPES: 257–79.

Imazeki, Jennifer. 2015. “Getting Students to Do Economics: An Introduction to Team-Based Learning.” *International Advances in Economic Research* 21 (4): 399–412.

Johnson, Bruce K., John J. Perry, and Marie Petkus. 2012. “The Status of Econometrics in the Economics Major: A Survey.” *Journal of Economic Education* 43 (3): 315–24.

Klein, Christopher C. 2013. “Econometrics as a Capstone Course in Economics.” *Journal of Economic Education* 44 (3): 268–76.

Knittel, Christopher R, and Konstantinos Metaxoglou. 2018. “Working with Data: Two Empiricists’ Experience.” *Journal of Econometric Methods*, no. 1 (January). De Gruyter: 1–12.

McGoldrick, KimMarie. 2008. “Doing Economics: Enhancing Skills Through a Process-Oriented Senior Research Course.” *The Journal of Economic Education* 39 (4): 342–56.

Open Science Collaboration. 2015. “Estimating the Reproducibility of Psychological Science.” *Science* 349 (6251). American Association for the Advancement of Science.

Pace, R. Kelley, and Otis W. Gilley. 1997. “Using the Spatial Configuration of the Data to Improve Estimation.” *Journal of Real Estate Finance and Economics* 14 (3): 333–40.

Pruim, Randall, Daniel T Kaplan, and Nicholas J Horton. 2017. “The Mosaic Package: Helping Students to ’Think with Data’ Using R.” *The R Journal* 9 (1): 77–102.

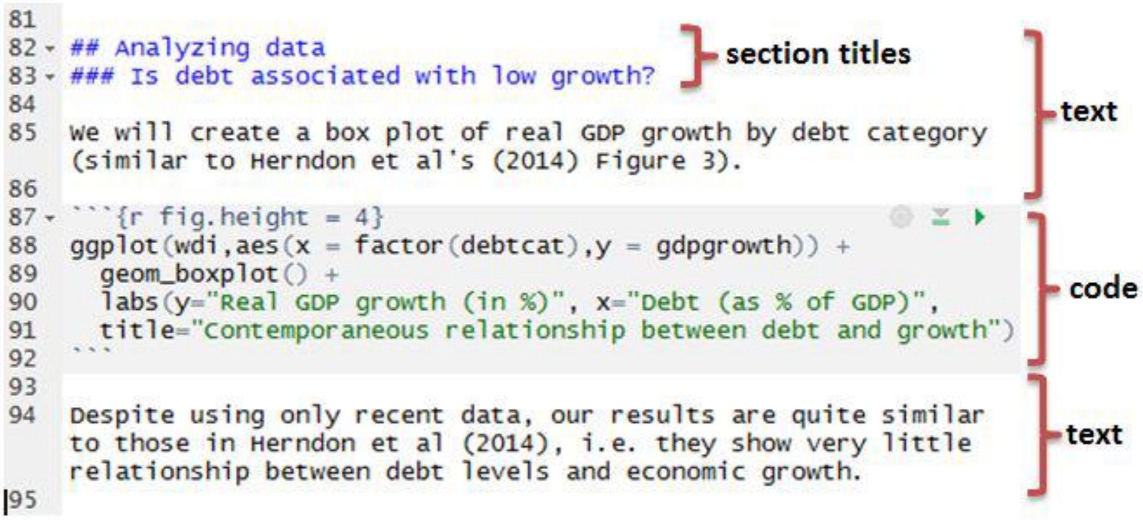
Reinhart, Carmen M, and Kenneth S Rogoff. 2010. “Growth in a Time of Debt (Digest Summary).” *American Economic Review* 100 (2). CFA Institute: 573–78.

Wegener, Rasmus, and Velu Sinha. 2013. “The Value of Big Data: How Analytics Differentiates Winners.” Bain; Company. http://aabdp.org/BAIN%20_BRIEF_The_value_of_Big_Data.pdf.

Wickham, Hadley. 2017. *Tidyverse: Easily Install and Load ’Tidyverse’ Packages*. https://CRAN.R-project.org/package=tidyverse.

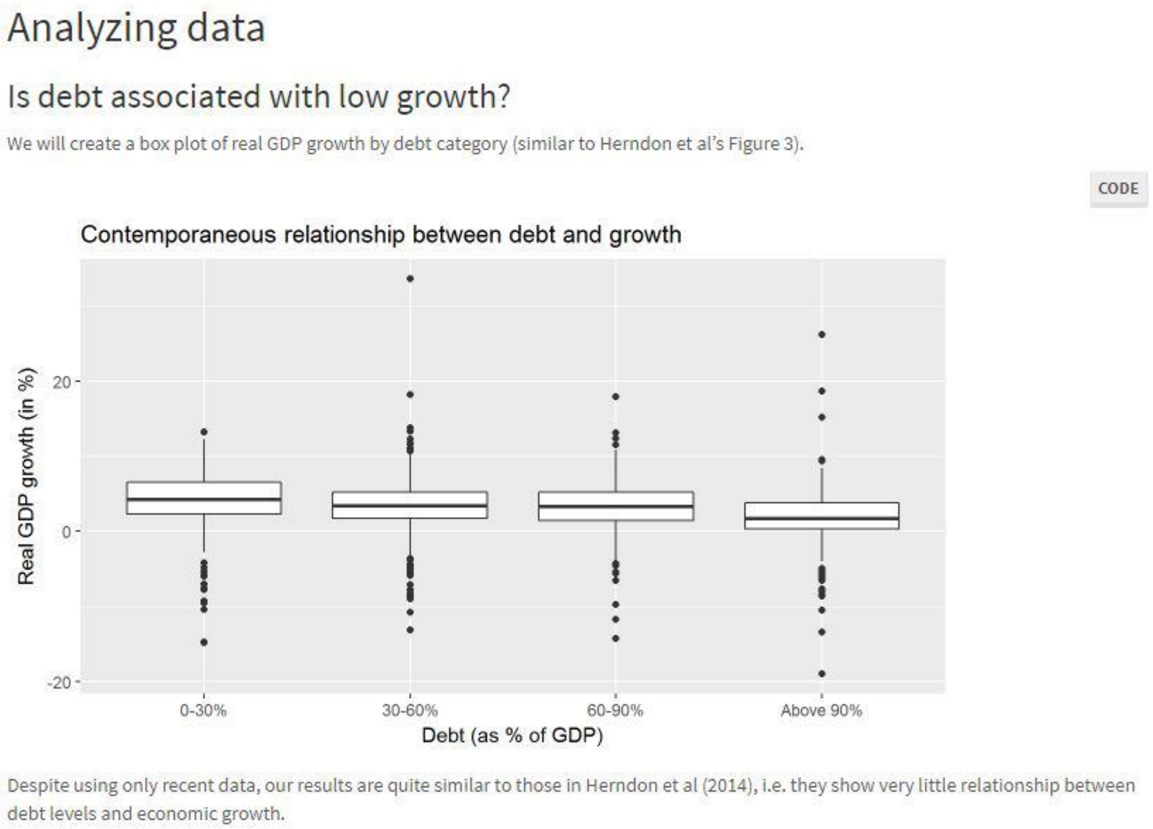
Table 1

Coding Empirical Analysis Competencies
Basic: Loading data Calculating descriptive statistics Estimating a regression model
Medium: Creating new variables Selecting observations/variables Creating a graph/visualization
Advanced: Merging/joining data Summarizing data by groups Reshaping data



~~~

Analyzing data
Is debt associated with low growth?

We will create a box plot of real GDP growth by debt category (similar to Herndon et al's (2014) Figure 3).

```{r fig.height = 4}
ggplot(wdi,aes(x = factor(debtcat),y = gdpgrowth)) +
  geom_boxplot() +
  labs(y="Real GDP growth (in %)", x="Debt (as % of GDP)",
  title="Contemporaneous relationship between debt and growth")
```

Despite using only recent data, our results are quite similar to those in Herndon et al (2014), i.e. they show very little relationship between debt levels and economic growth.

~~~

Figure 1: Example of a source R Markdown file



Analyzing data

Is debt associated with low growth?

We will create a box plot of real GDP growth by debt category (similar to Herndon et al's Figure 3).

Despite using only recent data, our results are quite similar to those in Herndon et al (2014), i.e. they show very little relationship between debt levels and economic growth.

Figure 2: Example of a knitted R Markdown document