Teaching Programming in Econometrics

Tomas Dvorak, Department of Economics
Union College, Schenectady, NY

Abstract: Over the last few years, three broad trends have emerged in the practice of econometrics. The first is the focus on research design and estimating causal effects as described in Angrist and Pischke (2010). The second trend is the use of big data as described by Einav and Levin (2014) and Varian (2014). The final trend is to make empirical research transparent and reproducible as described in Ball and Medeiros (2012). These trends raise demand for programming skills. Econometrics is no longer done using a point-and-click or copy-and-paste method. Instead, data retrieval, preparation, manipulation and analysis require programming in statistical software. Yet, undergraduate econometrics courses rarely explicitly teach students how to program. In this paper, I describe five programming skills needed in econometrics: data retrieval, selecting observations and variables, transforming variables, merging and appending data, and aggregating and reshaping data. I argue that these skills lead to more meaningful analyses by enabling students to combine and manipulate existing data as well as take advantage of new data. In addition, using statistical programming enables students to make their research transparent and reproducible.

1. Introduction

Programming statistical software is an important part of what economists do. Consider Table 1 below, which lists the eight most recent winners of the best paper awards for publications in the American Economic Association's two prestigious journals: *AEJ Applied Economics*, and *AEJ Economic Policy*. All of these papers present empirical evidence. Importantly, all but one of these papers posts their data and programs online. The papers use a mixture of data sets: from surveys following a field experiment (Dupas, 2011) to publicly available macroeconomic data (Auerbach, and Gorodnichenko, 2012); from data on the universe of prison inmates in Italy (Mastrobuoni and Pinotti, 2015) to administrative employment records from Canada (Oreopoulos, von Wachter and Heisz, 2012). One thing that the papers have in common is the use of programs (five used Stata, one used R, and one Matlab). The number of programs used in each paper ranges from 3 to 59, with a median of 7. Even the most straightforward analysis required data manipulation: selecting observations, creating new variables, and lots of merging and aggregating. Of course, the programs also included the analysis: commands for descriptive statistics, tables, graphs, regressions, etc. The median size of the programs needed for each paper is 55KB, which corresponds to about 1000 lines or 20 pages of code. Needless to say, many programs are longer than the papers themselves.[1]

[1] My highly selective sample of papers may overestimate the use of programming in economics. If that is the case, however, it shows that the profession values programming and the clever of identification strategies and skillful data manipulation that is associated with it. Perhaps collecting data on the use of programming for papers that did *not* make the best paper awards would be useful.

Table 1: Best Paper Award Winners AEJ: Applied Economics, AEJ: Economic Policy, 2016-2012

Citation	Title	Data	Empirical Strategy	Number of Programs KB of Code
Mastrobuoni and Pinotti, 2015	Legal Status and the Criminal Activity of Immigrants	universe of prison inmates	difference-in-difference	6 programs 34 KB
Gaynor, Moreno-Serra and Propper, 2013	Death by Market Power: Reform, Competition, and Patient Outcomes in the National Health Service	large number of administrative data, hospital admissions	difference-in-difference	14 programs 145 KB
Moretti, 2013	Real Wage Inequality	US Census, BLS CPI, ACCRA	measurement of inequality	6 programs 55 KB
Auerbach and Gorodnichenko, 2012	Measuring the Output Responses to Fiscal Policy.	NIPA, RSQE, SPF, Greenbook	structural VAR	59 programs 400 KB
Dupas, 2011	Do Teenagers Respond to HIV Risk Information? Evidence from a Field Experiment in Kenya	surveys including several follow up surveys	randomized trial, difference-in-difference	3 programs 41KB
Niehaus and Sukhtankar, 2013	Corruption Dynamics: The Golden Goose Effect.	Official work records, household survey	difference-in-difference	7 programs 111 KB
Oreopoulos, von Wachter and Heisz, 2012	The Short- and Long-Term Career Effects of Graduating in a Recession	administrative datasets from Statistics Canada	panel regression	not provided
Chodorow-Reich, Feiveson, Liscow and Gui Woolston, 2012	Does State Fiscal Relief during Recessions Increase Employment? Evidence from the American Recovery and Reinvestment Act	CES, FRED, BLS, Medicaid, ARRA	instrumental variable	10 programs 23 KB

There are three broad trends that drive the need for programming in economics. The first trend is the advances in research design. Described in Angrist and Pischke (2010), these advances include the use of experimental and quasi-experimental data. Half of the winners in Table 1 used difference-in-difference specifications using experimental (Dupas, 2013) or quasi-experimental data (Mastrobuoni and Pinotti, 2015; Gaynor et al, 2013; Gaynor et al, 2015). Although in principle straightforward, the implementation of these strategies requires considerable data manipulation and programming. For example, Gaynor et al (2015) required merging a variety of administrative data sets, matching patient level data with hospital level data, calculating market structure in various geographic regions, etc. Another popular quasi-experimental strategy is regression discontinuity (RD). As described by Imbens and Lemieux (2008), credible RD requires extensive plotting of the outcome variable, examination of

covariates around the discontinuity, and a number of sensitivity analyses. For example, Black (1999) identifies the value of better schools by comparing housing prices on the boundary of attendance districts. Identifying such houses requires skillful data collection and manipulation.

The second trend that raises the demand for programming in economics is the use of big data. Einav and Levin (2014) describe how large scale administrative data sets and private sector data will transform economic research. Working with big data requires programming skills. Varian (2014), in his article entitled "New Tricks for Econometrics," specifically points out the need for skills to retrieve and manipulate big data (e.g. via SQL). In the context of the undergraduate curriculum, the need for programming is probably even higher since most economics majors find employment in the private sector rather than pursuing a PhD in economics. Their private sector jobs are likely to require working with larger and more diverse data than those available to academic economists.

The final trend is the need for reproducible research as articulated by Ball and Medeiros (2012). The key to reproducible research is to faithfully record all data manipulations from downloading the raw data to producing tables and graphs. This is done with a computer program. Thus, without programming skills students cannot do reproducible research. Reproducibility is important not only to ensure integrity of research, but also to enable other researchers to build on existing work. Testing the sensitivity of results to a variety of samples and manipulations is only possible if a program is available. In fact, after challenging the credibility of empirical work in Leamer (1983), Leamer's response to Angrist and Pischke (2010) calls for sensitivity analyses (see Leamer, 2010). He says that without sensitivity analyses, and I would add without programs and data, it is like "like a court of law in which we hear only the experts on the plaintiff's side, but are wise enough to know that there are abundant arguments for the defense."

2. Programming skills are mostly absent from econometrics curricula

Despite its pervasiveness in the practice of econometrics, programming appears mostly absent in the econometrics curricula. Table 2 lists a number of leading undergraduate and graduate econometrics textbooks. The content of these textbooks focuses on econometric methods (hypothesis testing, properties of estimators, regression coefficients, etc.). With the exception of Christopher Baum's *An Introduction to Modern Econometrics Using Stata*, the textbooks contain very little programming. When they do have programming, it is usually one line of code to execute a particular method (e.g. regress y x1 x2). Most textbooks come with sample data, but this data is always highly processed and cleaned up. In other words, econometrics textbooks don't teach data retrieval and manipulation. They teach econometric methods.

Table 2: Leading Econometrics Textbooks

Title	Author	Programming Content
Panel A: Undergraduate Textbooks		
Real Econometrics	Michael A. Bailey	Computing corner: one line commands for Stata and R, discusses replication (p. 28)
Using Econometrics: A Practical Guide (6th ed)	A. H. Studenmund	no computer commands at all, chapter on "running your own regression project" (Chap 11). no programming
Basic Econometrics (4th ed)	Damorad N. Gujarati	no computer commands at all, no tips for implementing a project
Principles of Econometrics	R. Carter Hill, William E. Griffiths, Guay C. Lim	section on research process, supplementary materials for EViews, Stata and other packages are available, mostly using point and click and analysis of cleaned up data
Introduction to Econometrics	James H. Stock and Mark W. Watson	chapter on assessing empirical studies, data available but all data is processed and cleaned up, no specific software mentioned
Introductory Econometrics: A Modern Approach	Jeffrey M. Wooldridge	data in various formats, no commands, no manipulation, there exists supplementary text using R by Florian Heiss
Introduction to Econometrics (4th ed)	Christopher Dougherty	one line Stata commands for regressions, no chapter on projects or data manipulation
An Introduction to Modern Econometrics Using Stata	Christopher F. Baum	good amount programming, from reading data into Stata, merging, appending, even reshaping
Panel B: Graduate Textbooks		
Econometric Analysis of Cross Section and Panel Data (2nd ed)	Jeffrey M. Wooldridge	has link to Stata commands for executing the methods on processed data
Econometric Analysis (7th ed)	William H. Greene	none
Econometrics	Fumio Hayashi	none
Microeconometrics: Methods and Applications	A. Colin Cameron and Pravin K. Trivedi	none, but has a companion text for doing all examples in Stata

Three of the books have accompanying texts that provide implementation of examples. First, Wooldridge's undergraduate text has an accompanying book entitled *Using R for Introductory Econometrics*, published earlier this year by Florian Heiss. The book describes how to implement all of Wooldridge's examples in R. It is an incredibly useful resource that introduces students to basics of programming in R, including loading-in data, data types, etc. Second, Hill, Griffiths and Lim's book also has a set of accompanying texts for doing textbook examples in Stata, R, EViews and other packages. Finally, the graduate text by Cameron and Trivedi has the accompanying *Microeconometrics Using Stata* written by the authors themselves.

Perhaps the emerging model for the curriculum is using supplementary texts that include programming. Still, the emphasis in the current collection of accompanying books is working with the highly processed and cleaned up data. There is very little about retrieving, combining and manipulating data that had to be done to create the example data. In other words, the examples focus on estimation – the final step in the analysis – rather than on data retrieval and manipulation.

On the one hand, the general absence of formal programming instruction from econometrics textbooks is understandable. There is a lot of material to cover, even in an introductory econometrics course (sampling distribution, hypothesis testing, simple and multiple regression, non-linear models, panel data, limited dependent variable, instrumental variables). These are difficult concepts that require explanation and practice, even for students with a probability and statistics background.

On the other hand, a lack of programming skills as part of the econometrics curriculum limits what students can actually do with their econometrics skills. Most of them will be able to estimate models on cleaned up data. However, they will have difficulty putting together their own data, combining data sets and manipulating data in a reproducible way. For many economists, discovering new data, finding new combinations of data and manipulating data in new and interesting ways is the most exciting part of empirical work. Research design is the true potential of econometrics. My experience is that students are curious about new data. They are skillful at finding it, but lack the tools to manipulate it so that they can apply econometric methods to it. Methods are important, but methods without data is like a paintbrush without paint. I believe that at the graduate level, students are expected to learn programming on their own. Such expectation seems unreasonable at the undergraduate level. Therefore, in this essay I list the key statistical software programming skills that would enable students to unleash their creativity on the rich and vibrant colors of data that is all around us.

3. Programming Skills for Undergraduate Econometrics Curriculum

3.1 Data retrieval

One of the initial steps in any econometrics project is data retrieval. Most commonly, data is imported from a text or comma-delimited file stored on a local hard drive. This data should remain it its original form. Only the absolutely necessary alternations, such as unzipping the file, should be done prior to importing the data into statistical software. For example, deleting columns or rows should be done with code *after* the data is imported.

In many cases it is possible to download data directly into statistical software. This skips the step of downloading data onto a local drive. For example, both Stata and R can point their *insheet* or *read.csv* commands to a url address. Another way of directly importing data into statistical software is via commands that access certain databases. For example, both Stata and R have commands that access World Development Indicators (WDI). Stata's *freduse* gives access to St.Louis Fed's FRED. R's package *quandl* also gives direct access to FRED in addition to a large number of other data sources.

The advantage of importing data directly from a publicly available web address is that those who try to reproduce the analysis don't have to first download and save the data on a local drive. Updating the analysis is also straightforward: the call to the database or a url always retrieves the most up-to-date data. The downside of up-to-date data is that replication may use more recent data than the original analysis. Therefore, it is good practice to store directly downloaded data on a local drive or in a separate online location (e.g. dropbox), thus preserving the data as it was originally downloaded.

3.2 Selecting observations and variables

Importing raw data necessitates further manipulation. This could involve selecting particular observations (years, countries, individuals), dropping observations with missing values, etc. The code that selects the sample needs to be part of the record associated with the analysis. This is important for at least two reasons. First, running analyses on different samples is the hallmark of many robustness tests. Second, other researchers can easily modify the code and re-run the analysis as advocated in Leamer (2010).

The code should make using alternative variables easy. Databases often have multiple variables measuring the same concept. Reading-in all of the variables and then using the code to select alternatives makes sensitivity analyses possible. In an influential study, Easterly (2003) showed that a slight change in the definition of aid or good policy leads to opposite conclusions from previous studies.

3. 3. Transforming variables

An essential part of data manipulation is the creation of new variables as functions of existing variables (logs, percent changes, sums, etc.). In transforming variables, students need to pay attention to different data types and formats (strings, numbers, dates). Adding two numbers that are stored as strings does not work. Sorting by date which is stored as a string does not work either. Properly formatting the date variable is also essential for merging data with other data sets. Sometimes students may need to search for new functions. For example, my students were stumped with political donations data in which some zip codes were given as 5 digit and some as 9 digit. In order merge the data with the census data, they had to transform all zipcodes into 5-digit zipcodes. Similarly, working with lags and leads allowed them to create variety new useful variables.

Summarizing within groups involves creating a new variable that is a function of values of other variables within groups of observations. For example, in data on individuals we may want to calculate average wage within each state. Calculating data within groups is useful in creating shares. In firm level data, calculating a firm's share in its industry involves calculating totals for each industry. Calculations within groups can often be used to filter data. For example, in a panel data on countries, we may want to exclude countries that at any point had GPD below a certain threshold. This means calculating minimum GDP within each country.[2]

3.4 Merging and appending

[2] Stata's egen and R's group_by() and mutate() accomplish this.

Perhaps the most common data manipulation involves combining datasets. This could be merging data one-to-one, e.g. merging monthly returns on S&P500 with monthly data on the federal funds rate. Merging can also be many-to-one as when we merge data on individuals with information on the states they live in. The key for students to understand is that both data sets need to have at least one common variable by which they will be merged. I find it very useful to ask students how many observations the merged data set should have. Also, if there is not a perfect match, can they explain why? This type of work leads them to become deeply familiar with their data, and to avoid errors.

Appending is stacking two data sets with identical sets of variables on top of each other. This could arise when we want to combine several cross-sections from different years, or several time-series for different entities.

3.5 Aggregating and reshaping

A particularly useful data manipulation is aggregating data by groups. For instance, one of the AEJ winners, Oreopoulos et al (2012), collapsed data on individuals down to data on regions, year, and age category, so that it could be matched with data on unemployment which was only available by region, year and age category.

A particularly challenging manipulation is reshaping of data. Occasionally, some rows need to be turned into columns and vice versa. It is important for students to determine what the unit of observation is. For example, in a panel regression, the unit of observation may be year and state. If the data is stored such that rows are the years and columns are the states, students will need to reshape that data so that each row represents state and year. Also, reshaping is sometimes necessary for effective visualization. For example, making a time plot of the number of wins for each team requires each team to be a variable. Thus, being able to mold the data into the most appropriate shape is an important skill.

4. Two teaching examples

I created two online examples of econometrics programming for use in undergraduate econometrics courses. Both examples illustrate most of the concepts above. The econometric analyses are fairly simple – a multiple regression is the most sophisticated method – but the data preparation requires programming and manipulation skills. The first example uses Stata. The subject is debt and economic growth along the lines of a famous Reinhart and Rogoff (2010) study which was, in fact, done in excel and later shown to be incorrect. This affair provides excellent motivation for students to learn programming. The second example uses R. It is a follow up on my earlier work, Dvorak (2007), which created a sample annotated paper in econometrics. The sample paper showed students what an empirical paper should look like, with annotations describing the structure of each section and organization of tables. The sample paper, however, did not include information on how the data for it was put together. This programming example remedies that shortcoming.

5. Conclusion

Econometrics is the dominant methodology in economics. For example, Hammermesh (2013) finds that papers published in the top three economics journals have become more empirical. Also, in a survey of economics undergraduate curricula, Johnson, Perry and Petkus (2012) find that econometrics has become more common as a requirement for an economics major, especially at top rated schools. In this essay I argued that today's nature of empirical work in economics requires a great deal of programming, and therefore, that programming should be part of the undergraduate econometrics curriculum.

Introducing programming into undergraduate economics curricula will come at a cost. It is not clear which topics commonly covered in these courses deserve being replaced with material on programming and data manipulation. In the age of big data, topics that deal with small sample properties of estimators seem like good candidates to go. The key will be weighing the marginal value of teaching an additional econometric method against the marginal value of an additional programming and data manipulation. If we find our students are applying highly sophisticated techniques to uninteresting data, or that their studies cannot be replicated, we have probably gone too far teaching the methods.

I conclude with the adage that data preparation and manipulation accounts for 80% of empirical work, while the analysis takes only 20% (e.g. Lohr, 2014). In the current econometrics curriculum the balance is, at best, flipped the other way. This disparity deserves a conversation among those of us who teach econometrics. My hope is that this essay is a start to that conversation.

References:

Auerbach, Alan J. and Yuriy Gorodnichenko. 2012. "Measuring the Output Responses to Fiscal Policy." *American Economic Journal: Economic Policy*, 4(2): 1-27.

Ball, Richard, and Norm Medeiros. "Teaching Integrity in Empirical Research: A Protocol for Documenting Data Management and Analysis." *The Journal of Economic Education* 43, no. 2 (2012): 182-189.

Black, Sandra E. "Do better schools matter? Parental valuation of elementary education." Quarterly Journal of Economics (1999): 577-599.

Jonathan Hall 1 Cory Kendrick 2 Chris Nosko ,The Effects of Uber's Surge Pricing: A Case Study, http://faculty.chicagobooth.edu/chris.nosko/research/effects_of_uber's_surge_pricing.pdf

Blake, Thomas, Chris Nosko, and Steven Tadelis , "Consumer Heterogeneity and Paid Search Effectiveness: A Large Scale Field Experiment" with. Econometrica 83(1) (2015) pp 155-174.

Chodorow-Reich, Gabriel, Laura Feiveson, Zachary Liscow and William Gui Woolston. 2012. "Does State Fiscal Relief during Recessions Increase Employment? Evidence from the American Recovery and Reinvestment Act."*American Economic Journal: Economic Policy*, 4(3): 118-45.

Dupas, Pascaline. 2011. "Do Teenagers Respond to HIV Risk Information? Evidence from a Field Experiment in Kenya." *American Economic Journal: Applied Economics*, 3(1): 1-34.

Dvorak, Tomas. "An Annotated Sample Paper in Econometrics." *The Journal of Economic Education* 38, no. 1 (2007): 124-124.

Easterly, William, 2003, Can Aid Buy Growth?, Journal of Economic Perspectives—Volume 17, Number 3—Summer 2003—Pages 23–48.

Einav, Liran, and Jonathan Levin. "Economics in the age of big data."*Science* 346, no. 6210 (2014): 1243089.

Einav, Liran, and Jonathan Levin , The Data Revolution and Economic Analysis, *Innovation Policy and the Economy*, Volume 14, edited by Josh Lerner and Scott Stern, May 2014, 1-24

Gaynor, Martin, Rodrigo Moreno-Serra and Carol Propper. 2013. "Death by Market Power: Reform, Competition, and Patient Outcomes in the National Health Service." *American Economic Journal: Economic Policy*, 5(4): 134-66.

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

Imbens, Guido W., Thomas Lemieux Journal of Econometrics 142 (2008) 615–635 Regression discontinuity designs: A guide to practice,

Johnson, Bruce K., John J. Perry, and Marie Petkus , 2012, The Status of Econometrics in the Economics Major: A Survey, THE JOURNAL OF ECONOMIC EDUCATION, 43(3), 315–324, 2012

Kearney, Melissa S., and Phillip B. Levine. "Media Influences on Social Outcomes: The Impact of MTV's 16 and Pregnant on Teen Childbearing."*The American Economic Review* 105, no. 12 (2015): 3597-3632.

Leamer, Edward E. "Let's Take the Con Out of Econometrics." The American Economic Review 73, no. 1 (1983): 31-43. http://www.jstor.org/stable/1803924.

Leamer, Edward E. 2010. "Tantalus on the Road to Asymptopia." *Journal of Economic Perspectives*, 24(2): 31-46.

Lohr, Steve, 2014, For Big-Data Scientists, 'Janitor Work' Is Key Hurdle to Insights, New York Times, Aug 17, 2014, http://nyti.ms/1t8IzfE

Mastrobuoni, Giovanni and Paolo Pinotti. 2015. "Legal Status and the Criminal Activity of Immigrants." *American Economic Journal: Applied Economics*, 7(2): 175-206.

Moretti, Enrico. 2013. "Real Wage Inequality." *American Economic Journal: Applied Economics*, 5(1): 65-103.

Niehaus, Paul and Sandip Sukhtankar. 2013. "Corruption Dynamics: The Golden Goose Effect." *American Economic Journal: Economic Policy*, 5(4): 230-69.

Oreopoulos, Philip, Till von Wachter and Andrew Heisz. 2012. "The Short- and Long-Term Career Effects of Graduating in a Recession." *American Economic Journal: Applied Economics*, 4(1): 1-29.

Rogoff, Kenneth, and Carmen Reinhart. "Growth in a Time of Debt."*American Economic Review* 100, no. 2 (2010): 573-8.