Which statistical software to use?

A reflection and review upon the use of STATA, R and Python for teaching in the social sciences

Thomas de Graaff[1]*

Abstract

Keywords
Teaching—data—empirical research—STATA—R—Python

[1] *Department of Spatial Economics, Vrije Universiteit Amsterdam, Amsterdam, The Netherlands*
***Corresponding author**: ✉ t.de.graaff@vu.nl; thomasdegraaff.nl

Introduction: the empirical workflow

> Econometrics is much easier without the data.
>
> Marno Verbeek

The quote above does not only apply to economics and econometrics, but to all of the social sciences in general. Empirical research—that is, dealing with data in all its forms—requires a rigorous approach, even more so, with the increasing emphasis on openness and reproducibility of all kinds of scientific research. Therefore, it is strange that in academic education there is not much guidance in choosing which tools to use and in the philosophy behing choosing an efficient and reproducable workflow.[1]

This note deals with the suitability of various software packages for applying applied econometrics in specific and data science in general.[2] Specifically, In will focus on STATA, R and Python.[3] I will not say which tool to use. Instead, I will focus on the various strengths and weaknessess of each software package combined with it specific approach. The main criteria I will consider are the package suitability for education and how well it can be integrated in an efficient workflow. The former is mostly important for doing (small) data exercises, whilst the latter is vital for larger research project, such as theses and later on perhaps research papers.

To illustrate why the concept of reproducibility is vital, note that a typical empirical workflow in the social sciences looks as follows:

Generate data Data is read from an external source (file or online database) or is simulated.

Manipulating data This is usually the most time demanding phase[4] and includes (amongst many other things) manipulating missing data, merging data and relabeling data

Analyse data This phase includes not only standard ecometrics and statistical or machine learning techniques, but as well as graphical representations as maps and figures.

Present results Finally, documents in the forms of papers, posters, theses, or presentations have to be drafted. Note that ideally one wants to do so in various formats, such as in pdf for physical papers and in html for webdisplay.

Unfortunately, all these steps do not necessarily run sequentially in time. Supervisors, referees, colleagues, and the future you, always want to add or delete elements to or from your research. For instance, variables have to be added, models specifications have to be checked, and 3D pie charts have to be changed in something useful.[5]

Therefore, it is vital that all these steps are both (*i*) very well documented so that the future you can easily retrace your steps, implement changes and redo the whole research if needed, and (*ii*) well connected to each other. The latter does not necessarily entail that the whole research should be done in one software environment, but instead that the outcome of one research step (e.g., generating data) can easily serve as an input for another step (e.g., data manipulation).

In the next section I will lay out the strengths and weaknesses of three statistical software packages with this workflow in mind.

[1] There are some exceptions, see, e.g., Healy (2011) and Arribas-Bel and de Graaff (2014) for a workshop I gave together with Daniel Arribas-Bel.

[2] There is a difference between econometrics on the one hand and statistics on the other hand. Economics students first and foremost need to able to apply applied econometric techniques, such as presented in Angrist and Pischke (2008)

[3] I will also briefly tough upon some other packages, but these three mentioned are most likely the ones most used in economics, except of course for the ubiquitous Excel.

[4] There is and old saying that says that 80% of your research time goes in transforming data, while 20% is only spent on analysing the data.

[5] For a wonderful timelapse video on the nonlinearity and even sometimes chaos of writing a research paper, see this link.

Statistical software packages

I review the various packages according to several criteria. There are several others that will be discussed, but these I find most important for a suitable software package to be used for teaching in the social sciences.

Open source The most important argument to use an open source package is reproducibility. Your work is simply less accessible and thus reproducible if the code can only be run with applications that costs over €1,000.

Learning curve First, and foremost, students should be able to use the package for straightforward econometric research. If that is not possible after one six-week's course, the software package is not particularly suitable.

Size of the community Nobody wants to be locked in with obsolete technology. A large userbase ensures a high probability that the software package will be used and maintained in the future as well. Moreover, all sorts of indirect effects, such as user written routines, packages and documentation, come along for free with a large community.

Usefulness outside academia Often forgotten as an argument but outside academic life, some applications are more used than others. And with the recent emphasis on better preparation for the labor market, this argument seems to become more important. By the way, the application still mostly used would be the ubiquitous Excel and its related visual basic language.

Flexibility Ideally, a software approach should be both extendable and scalable. The former ensures that slight deviations from standard approaches can relatively easy be implemented. The latter is important when the size of the database increases, as typically is the case with recent improvements in remote sensing techniques.

Scriptable Finally, a software package should be scriptable—both internally as externally. Internally scriptable indicates that within the package scripts or programs can be written so that every step within the workflow can be reproduced. With externally scriptable I mean that the software package should also be used in combination with other software packages or languages, such as LaTeX, markdown, make, html, sql, C++, etc.

Other software packages that I will not review, typically score badly on more than one of these criteria. For example, SPSS is proprietary software, which has some issues with scriptability and is not very flexible. Moreover, a sizeable part of the user community moved over to other software packages (most notably R). Other software packages I will not review for reasons of similarity (such as Matlab to some extent) or my lack of experience with them (most notably SAS).[6]

STATA

STATA is proprietary software copyrighted by the StataCorp LLC corporation.[7] It is especially popular among economists, although the growth of STATA seems almost comparable with that of R (Python is in its own league, however).[8]

The biggest comparative advantage of STATA is its learning curve (although some students might disagree). It is relatively straightforward to teach students basic econometrics (including time series, panel data and count data techniques, discrete choice models and duration models). Moreover, they can do so in a structured way by writing scripts (the so-called do files). Many students appreciate as well the fact that STATA command can be given interactively (via a drop-down menu) of directly from the console enabling them to memorize the various commands on the fly.

There is a large STATA community. This ensures the availability of many useful so-called user-written routines and tutorial material, whether via Youtube or in pdf. The STATA community is not only huge, but as well very helpful (although the help documentation is oftentimes cumbersome). The first hit on Google on a sometimes not very focused question usually suffices.

Unfortunately, not many organizations outside academia use STATA—although I heard recently that some Danish consultancy agencies in Denmark do. Most organizations now typically use R or Python in combination with even more focused software applications as Hadoop or Julia (again, apart from Excel). Usually, this does not matter much as programming skills are easily transferable, but, unfortunately, the STATA language is hardly a programming language. Instead, it is more a sequence of very specific commands. Therefore, it is hard to relate the commands to something as Python.

Another downside is the flexibility of STATA. For procedures that slightly deviate from the 'basic' ones[9], STATA can give you a hard time. Mostly, because the source code is not known, working with matrices is cumbersome (to say the least) and the number of programming tools is rather limited. Moreover, STATA can only work with one active dataset, which makes merging datasets sometimes difficult. Finally, although there are some plugins, STATA is not designed for working with data stored on servers somewhere else via application programming interfaces (API's). Unfortu-

[6] I should mention the object oriented matrix language Ox as well, which is mostly used by econometricians. Because of the steep learning curve and relatively small community I will not consider it here, but I am aware of its popularity in some particular research groups.

[7] See https://www.stata.com/.

[8] I have to be careful though for these sorts of statements. Usually, the popularity of software applications is researched by looking at search engine counts for jobs, frequently asked questions or counting popularity on the stackoverflow site (https://stackoverflow.com/). An interesting recent overview can be found on http://r4stats.com/2017/06/19/scholarly-articles/.

[9] Although the set of 'basic' procedures is quite extensive.

nately, working via API's[10] becomes increasingly important, if not only for the recent surge in open access data via local governments, Google, Twitter, Foursquare, etc.

STATA itself is well scriptable internally. although it is less programming and more sequencing commands, for the majority of research projects this is definitely good enough. Externally, however, it is another issues which has to do with its proprietary nature. Other applications, such as great ?nix utilities as make and pandoc but as well git and github have difficulties 'communicating' with STATA. It is possible, however, to export from STATA to other formats, especially the ubiquitous text files, which enables automatic generation of LaTeX tables.

Finally, STATA is alas not open source. This hinders reproducibility, insight in the source code ("How the hell did they do that?") as well as the transferability of data (the STATA .dta dataformat is not accessible to read directly, although many other packages, including R, have created special read procedures for STATA dataformats).

R

R emerged in 1995 as the open-source (GNU) version of the S language and quickly surpassed its predecessor in popularity.[11] R itself consists of a base distribution which can be enhanced with packages, and it is exactly the ease of writing, distributing and using packages that makes R so attractive for many users. At the moment there are more than 10,000 packages on the official CRAN website, but Github contains many more.[12]

Because of its open-source nature and its large and very active community, the quality of most of the packages is almost guaranteed. Indeed, if there are killer applications in the data science world, then R has most of them, with as most notifiable examples dplyr for quick and robust data manipulation and ggplot2 for structural plotting using the *grammar of graphics* approach as advocated by Wilkinson (2006) and implemented by Wickham and Chang (2013).

The learning curve of R is however steeper than that of STATA.[13] Moreover, R can be characterized as:

> first, R is written by statisticians, for statisticians.

Unfortunately, this can be seen in the R language itself. Best described as 'quirky', the R language is not the most beautiful or best designed programming language. For example, usually there is more than one (often actually more than three) ways to accomplish something, which leads to a myriad of styles and types of coding. In applied statistics though, R shines. Although slightly more cumbersome than STATA, commands for basic econometric techniques are relatively straightforward. Estimating a linear regression simply looks as follows:

```
# ordinary least squares regression
model <- lm(y ~ x1 + x2, data = data1)
```

Note the difference with STATA, where as in STATA one simply uses `reg y x1 x2`. However, in R one can store everything in an object (in this case the object "model") and use it later, and one can use several databases ('dataframes') at the same time. Both features are extremely useful for dealing with larger research projects. And although STATA still is more straightforward in basic econometrics, R users have developed several packages to emulate STATA's ease of use.[14]

Because R is very flexible and the community is that huge (across disciplines as well), for almost every (statistical) problem a package most likely is developed (if feasible). The problem is, however, that terminology differs across fields. Moreover, what economists find important (causality anyone?) is generally less important within biometrics, physics, statistics of sociology. So, using R also means spending a great deal of time finding and checking the correct package.[15]

If anything, R if flexible in terms of what it can do. Spatial data analysis, Bayesian inference, optimization, network analysis and—above all—producing wonderful plots that can even be used as interactive web applications. Therefore, it is used intensively outside academia as well by large companies such as Facebook, Google, Twitter, Microsoft, Uber and Airbnb, mainly for 'quick and dirty' data science and data and output visualization.

Unfortunately, R is not very scalable, as only internal computer memory (RAM) is used for data storage (similar to matlab, STATA and comparable languages). There are ways to solve this (typically in combination with other languages, such as Hadoop or using parallel computing techniques), but it is cumbersome for very large research projects with huge amounts of data (remote sensing data in geographical analysis comes to mind or up to 100 million records when using micro data).

Finally, R is extremely scriptable, both internally and externally. Internally, the R language is a full blow object-oriented language, although perhaps not the most beautiful one. Especially the ease of writing and reading packages is very useful for groups of researchers who are working on the same project. Externally, R really has a comparative advantage as that it can be scripted from a command line (a terminal), can be used in combination with other languages (e.g., python via rpy and C++ via rcpp to speed up procedures if needed) and works as a charm in combination with both

[10] Note that I do hesitate to drop the hype term 'big data' here.

[11] See https://www.r-project.org/about.html.

[12] See http://blog.revolutionanalytics.com/2017/01/cran-10000.html.

[13] Steep learning curves are in principle not problematic as it indicates that you learn rapidly and in the end the pay-off should be large. For 'quick and dirty' solutions, however, steep learning curves are problematic.

[14] Good examples are the margins package in R that has similar features as the very useful margins command in STATA and the plm package for implementing fixed effects.

[15] Fortunately, there is the great website CRAN Task Views which gives good overviews of the most important packages in several subfields, such as dealing with spatial data, Bayesian inference, econometrics and mathematical programming.

LaTeX and html for creating webpages.[16]

Python

Opposite to STATA and even R, Python is a general object oriented programming language more in line with a language such as C++.

Statistics in education

Concluding remarks

References

Angrist, J. D. and J.-S. Pischke (2008). *Mostly harmless econometrics: An empiricist's companion*. Princeton university press.

Arribas-Bel, D. and T. de Graaff (2014). *Workshops on open workflows, 2nd Edition*.

Healy, K. (2011). "Choosing your workflow applications". In: *The Political Methodologist* 18.2, pp. 9–18.

Knuth, D. E. (1984). "Literate programming". In: *The Computer Journal* 27.2, pp. 97–111.

Wickham, H. and W. Chang (2013). "An implementation of the Grammar of Graphics". In: *R package version*.

Wilkinson, L. (2006). *The grammar of graphics*. Springer Science & Business Media.

[16] R has one of the best implementations of so-called literate programming via the package knitr. Literate programming as originally coined by Knuth (1984) indicates that output of coding and documentation are generated simultaneously. Whether or not this is a good idea for larger research projects, literate programming can be very useful for generating documentation of R packages, web blogs and smaller research papers.