Econometric Software: A User's View

Jeffrey K. MacKie-Mason

The Journal of Economic Perspectives, Volume 6, Issue 4 (Autumn, 1992), 165-187.

Stable URL:
http://links.jstor.org/sici?sici=0895-3309%28199223%296%3A4%3C165%3AESAUV%3E2.0.CO%3B2-U

Journal of Economic Perspectives—Volume 6, Number 4—Fall 1992—Pages 165–187

Econometric Software: A User's View

Jeffrey K. MacKie-Mason

For the social scientist, software is a tool, not an end in itself. My objective in this review is to help practicing economists decide which tool will best get the task done. To do this I report on how seven programs performed on each of six different econometric projects. Comments on other aspects of the programs are sprinkled throughout.

Readers of this journal include research scholars, teachers, government and private sector forecasters and analysts, consultants, and others. In attempting to speak to all of these groups, I have tried to represent the viewpoint of a "serious but occasional" applied econometrician. "Occasional" because the person continually doing econometrics already knows what is available and has developed strong preferences. By "serious" I mean someone who cares to use a method appropriate to the problem at hand. Such a person is motivated to learn at least some simple programming—but usually not to write estimators from scratch!—and to want a tool that may sacrifice some ease-of-use for breadth and depth of econometric capabilities. It isn't possible to avoid some programming if you want to use the full range of tools taught in a first-year graduate sequence: no program has a complete enough command set.[1]

[1]Programming may mean little more than writing out a likelihood function, or retrieving the covariance matrices after two estimation runs and doing some simple algebra to compute a specification test. One program that does quite well without much programming is *Shazam*, which is widely used in teaching undergraduate and graduate courses.

■ *Jeffrey K. MacKie-Mason is Associate Professor of Economics and Public Policy, University of Michigan, Ann Arbor, Michigan, and Faculty Research Fellow, National Bureau of Economic Research, Cambridge, Massachusetts.*

Table 1
Software Reviewed and Hardware Platforms Supported by the Vendor

Product	*Platforms supported*[a]
Gauss[b]	MSDOS
Limdep	MSDOS, minicomputers, mainframes, UNIX workstations
RATS[b]	MSDOS, Macintosh, mainframes, minicomputers, UNIX workstations
SAS	MSDOS, Macintosh, mainframes, minicomputers, UNIX workstations
SST	MSDOS, UNIX workstations
Stata[b]	MSDOS, UNIX workstations
TSP	MSDOS, Macintosh, mainframes, minicomputers, UNIX workstations

[a]"UNIX" means that the program runs on some UNIX platforms. Most run full versions on only a few workstations, except *SAS* which runs on almost anything.
[b]The original draft of the review was based on the prior version of *RATS*, *Stata*, and *Gauss*. I updated the review, but did not work nearly as much with the new releases.

Programs, Platforms, Prices

Since I wanted to provide a user's experiences in some depth, I had to severely limit the number of software programs considered. I focus on six programs: *Limdep*, *RATS*, *SAS*, *Stata*, *SST*, and *TSP*. I more briefly review *Gauss*, which is a rather different product. Table 1 lists the versions of the programs reviewed.

To choose, I relied in part on the market to indicate the programs most used by applied econometricians. I also selected programs to meet some technological objectives: 1) they all run on IBM-compatible personal computers; 2) they all (with one temporary exception) run on at least one other, more powerful platform (mainframe or UNIX workstation); and 3) they all include some ability to program estimators and test statistics that are not provided as standard features.

Availability on different computer platforms (PC, mainframe, workstation) provides several advantages. Even if most work is done on the desktop, research assistants and co-authors at different locations may find it easier to use a mainframe or workstation. And, of course, choosing software from a company that is committed to more than one type of hardware provides some insurance in a world where computing technology continues to advance rapidly. This argument especially favors choosing a program that runs under UNIX as well as DOS, since it appears that research economists are beginning to move towards workstations as the platform of choice.[2]

[2]The UNIX workstation speed advantage can be large even for low-end machines. For comparison I simulated a regression dataset with two explanatory variables and 1000 observations, ran the regression, and repeated this 100 times. On a sample program with random number generation and linear regressions, *TSP* took 4.8 times as long as on a '386-33 mHz PC (with a '387 numeric coprocessor) as on a NeXT 33 mHz workstation. For *SAS* I ran my data handling test program

Table 2
Academic Prices for the Program Configurations Used in this Review

Product	*Price*	*Notes*
Gauss	$1135	Gauss386 3.0 plus Stats, Quantal, Max. Likelihood, TSCS, Linear Reg.
Limdep	$595	PC386, Version 6
RATS	$420	RATS386, Version 4.0
SAS[a]	$808	Base, Stat, ETS, IML, Version 6.04 (manuals not included)
SST	$500	Version 2
Stata	$245	Stata 3.0 Intercooled (386 Version)
TSP	$400	386TSP, Version 4.2A

[a]The *SAS* price is for the first year. The license must be renewed annually, for about 60 percent of the first year price.

I should mention that the choice of programs was not dictated by my personal use. *TSP* was the first program I learned of those reviewed here. In the intervening years, I've used *RATS* just as much, and both *SAS* and *Gauss* far more. Before preparing this article, I had not used *Limdep*, *SST*, or *Stata*.

A comparison of products for economists would be strange without price information. Unfortunately, it is quite difficult to provide prices concisely. Several of the programs have multiple configurations and modules; most of the programs charge different prices for different hardware platforms. Site licenses are often available. Students can get discounts. Table 2 presents list (academic) prices for PC versions of the bundles that I used for the review.

Method of the Review

Before describing the factors I did consider about each program, I should explain why I largely ignore two prominent features: user interface and graphics.

By user interface I mean the screens and menus seen by interactive users. Although graphical user interfaces are beginning to dominate the screens of desktop computers, they are not a prominent feature in serious econometric software. Of the programs below, the few alternatives to command line (or batch file) interfaces are mostly crude menu systems. *SAS* is an exception, but the window interface for *SAS* is awkward and hard to work—I've never met a *SAS* user who particularly likes it.

(described below) on both the PC and the NeXT; the NeXT version was 15 times faster. As it happens, PC-SAS makes exceptionally inefficient use of extended memory on the PC, so anything using large data files can be extraordinarily slow.

The lack of sophisticated graphical user interfaces is not surprising, since the interactive work for which they are best-suited is inappropriate for most econometrics. The applied econometrician is *constantly* redoing work after finding an error, a process which requires that empirical research be carefully documented and replicable. Even if one has a logging file open that captures all commands issued interactively through a graphical interface, it makes a very messy, hard-to-replicate record. Nearly all regular practitioners I know prefer to edit command files, then submit them all at once to the software. It's also a good habit for research assistants to learn since one can then easily maintain a "lab" notebook.

I do not consider graphics because all of the programs discussed here have easy-to-use, adequate facilities for simple scatter plots and histograms, which are useful for diagnosing residuals, the stationarity of time-series data, and so on. Plotting every data series is a good way to search for coding errors, but more graphics capability is needed rarely in econometric research. When presentation is important, most people with a desktop computer have at least one good option: a high-end spreadsheet program like Borland's Quattro Pro. However, if you use graphics frequently during your analysis, or want some specialized graph formats, be warned that the programs below *do* differ significantly in graphical capability.

In considering how to compare the programs, I quickly abandoned the idea of setting out formal criteria and assigning quantitative measures, leading to a comparison table with summary scores. Econometric tasks are simply too varied, and the criteria for good tools too subjective, for such a pseudo-quantitative analysis to be useful. Instead I present a more subjective, "user's view" that attempts to inform and provide guidance on issues like learning costs, data handling, programming ease, procedural breadth and depth, and documentation.

Rather than try to exhaustively compare capabilities, I prepared six econometric problems, using real data with the usual warts. Most of the remainder of this review will be structured around my experience in tackling these econometric problems with each program.[3] The six are:

1. Data Handling: construction of a large panel data set.[4] This project had five parts: read in firm data from COMPUSTAT (a database with corporate income statement and balance sheet data); select variables and observations

[3] I tried to limit jargon and methods discussed to those familiar from a first-year graduate sequence (the Cox model is an exception), but to aid in accessibility I occasionally provide a reference to a standard textbook treatment of the methods being discussed.

[4] Data handling features can make or break a program. A large fraction of the time on any project is spent preparing and manipulating the data. Further, given the lack of replication in economics, allowing errors to creep into the data should be a mortal sin. For these reasons, an appendix available from the author contains a copy of the code I wrote for each program to perform the data set construction task. Comparing the programs helps to give a realistic idea of how hard one must work to construct a dataset in each case, as well as some sense of the clarity and efficiency of the program's syntax.

from the firm data based on number of observations per firm and whether fiscal year was calendar year; convert missing data to the built-in missing code; create some new variables; and, read in some tax data then merge it with the firm data.[5]

2. *Panel Estimation:* estimation of a linear model using ordinary least squares, fixed effects, and random effects specifications, followed by specification testing (Judge et al., 1985, ch. 13);

3. *Simultaneous Equations:* estimation of a linear simultaneous equation system with suspected heteroskedasticity of unknown form (Judge et al., 1985, ch. 15);

4. *Nonlinear Least Squares:* estimation of a single equation by nonlinear least squares (Judge et al., 1985, ch. 6);

5. *Limited Dependent Variables:* estimation of a Cox proportional hazards model using data with censored observations. The Cox model is popular among labor economists for modeling the duration of events like unemployment spells (Amemiya, 1985, ch. 11; Judge et al., 1985, ch. 18). All of the programs handle probits and logits with aplomb, so there was little point in using these better-known models for the test.

6. *Hypothesis Testing:* testing linear and nonlinear hypotheses on the coefficients of a single equation (Judge et al., 1985, ch. 5.7).

The reader may notice a definite bias towards cross-sectional analyses in my choice of problems. This reflects, in part, my own research experience. In addition, focusing on the cross-section helps in limiting the number of programs viewed, thus allowing some depth in the reviews. When time-series analysis goes beyond classical serial correlation corrections into issues like Kalman filtering, spectral analysis, integrated and co-integrated processes, then some additional programs become major players. I comment briefly on the time-series capabilities of the programs below, but I do not push hard in this direction.

I now present the reviews of each program. The first six are presented in alphabetical order, followed by *Gauss*.

Limdep

Limdep began life specialized to limited dependent variable models. The author seems to have a never-ending interest in adding new capabilities, and in its current version it probably has the longest list of econometric models that

[5]If you habitually use more than one econometrics program on a PC, I urge you to obtain either Stat/Transfer or DBMS/COPY. These inexpensive programs quickly and painlessly translate data among several standard proprietary formats. The latter translates more formats, but most programs support a standard data interchange format; usually either dif, Lotus, or dBase. Of the programs reviewed here, *Stata* and *Gauss* do not support any interchange formats, but they are supported by both translation programs so I was able to move data easily among all six programs.

can be estimated with built-in commands. The author is particularly responsive to technical support questions and suggestions.

Limdep has some rough edges. For example it has no pure batch mode on the PC: output is always displayed on the screen. For runs with lots of output this is a problem—one program I used to compare speeds took 17.25 minutes versus 1.5 minutes for *TSP*, mostly because of the time *Limdep* spent writing to the screen. (This is an extreme case not meant to indicate the usual speed of *Limdep*. On my UNIX workstation, for which *Limdep* does have a pure batch mode, it took only twice as long as *TSP*.) As another example, when *Limdep* writes out data in a text file (say, for transfer to another program), it uses a format that sometimes leaves no spaces between two consecutive numbers so that they can only be read by something that specifies FORTRAN-style (fixed column) formats, not the more convenient free format. However, for many users these annoyances may be small compared to the value of a program that has built-in methods for so many different problems.

Data Handling. *Limdep* is fairly good at data handling, but not complete. I was able to handle the construction of my panel data set, although it required some clunky steps. For example, *Limdep* has no merge facility, but as long as the data are being matched by only one indicator variable (the year, in my case) it is possible to paste together a merge by sorting and using indexed vectors to match up the observations. Instead of a single merge command, though, this took more than a half dozen steps.

There are several other weaknesses. For example, missing values are stored as −999, which the estimation commands then treat as real data, so the user must remember to use a command that purges observations with missing values. Free-format input data cannot have lines longer than 300 characters. Text data are not permitted; since my data source uses character data to identify firms, I had to convert this data outside of *Limdep* before I could use it. The manual is honest about some limitations: "If you have a very large data set [sorting] is likely to be difficult or impossible to do. If so, you have probably not done enough processing of these data before using *Limdep*" (p. 122).

Panel Data. *Limdep* is superb for panel data. It easily handled my task, and the standard output included the most complete set of diagnostic and specification test statistics of any program. Beyond that, *Limdep* offers the greatest number of other panel data models for econometric analysis. One can add a time effect in addition to a cross-sectional (individual) effect. There are commands to estimate discrete choice models with panel data (fixed effects logit; random effects probit; conditional logit), and others. And for nearly all of these it is not necessary to have a balanced panel (same number of time series observations for each cross-sectional unit). *Limdep* also has the best data handling tools specifically designed for setting up panel data sets with various structures.

Simultaneous Equations. I had no problem with the estimation part of my task. But there is no provision for heteroskedasticity-consistent standard errors;

a user could program them but that is not a simple matter.[6] Overall, *Limdep* is rather thin on systems methods. For example, there is no full-information maximum likelihood estimator for a system, nor are there any commands to estimate a nonlinear system of equations.

Nonlinear Least Squares. It was easy to use the single-equation NLSQ command for my problem. I didn't need them this time, but nonlinear 2SLS and generalized method-of-moments estimation (which uses the optimal weighting of the instruments) are also available. Although several methods are provided, I felt that I obtained a rather incomplete analysis because of weak support for diagnostics and test statistics. Tests for serial correlation and heteroskedasticity are not built-in. Robust standard errors are not available.

Limited Dependent Variables. *Limdep* really shines in limited dependent variable analysis. The list of built-in methods is staggering, including the Cox proportional hazards model that I needed for my problem. One can manage most or all of these methods in other programs like *TSP* or *SST* by programming the appropriate likelihood function, but built-in commands are easier and less error prone. *Limdep* also offers matrix algebra and programming tools that are comparable to those in the other programs (other than *Gauss*), and dozens of application examples in the manual for extending the estimation and testing capabilities even further.

Nested logit is another good example (for example, Maddala 1983, section 3.6). This is an important method for estimating choice models with more than two choices. *Limdep* shows how to use the built-up commands to get consistent estimates, and is the only program that has an option to get correct (consistent) standard errors. The manual provides an extensive discussion of the theory behind the model, an elaborate example, and also the commands necessary to perform specification tests on the fundamental Independence of Irrelevant Alternatives assumptions that are the raison d'être for the nested logit model.

Hypothesis Testing. *Limdep* is mostly strong on hypothesis testing (but see "nonlinear least squares estimation" above). I could easily test my linear and nonlinear coefficient restrictions. Model specification tests are emphasized throughout the manual, with detailed examples showing how to apply them to nearly every model *Limdep* can estimate. The popular Breusch-Pagan test for heteroskedasticity is automatic for linear regression, and the methods for calculating other popular tests are shown in the manual and provided as sample programs with the software (see Judge et al., pp. 445–454). Several autocorrelation tests are provided or described.

[6] I found one troubling feature here. I tried requesting robust standard errors with the same option that had worked for single-equation ordinary least squares. In fact, robust standard errors are not available for simultaneous equations. Unfortunately, *Limdep* simply ignored my request without any warning, then reported the conventional standard errors. Thus, from the output file it appears that the request has been fulfilled, which could easily mislead a user. In at least one other place I requested a nonexistent option and got the same misleading result.

Ease of Learning and Documentation. The manual is easy to love and hate. It matches *Stata* and approaches *SAS* for completeness and generosity in providing numerous detailed examples. It compares favorably to the *TSP* manual for its discussions of the econometric theory underlying the methods. (*Limdep*'s author, William Greene, is also the author of a recent econometrics text.) Unfortunately, the manual is not well organized and much of the material is presented poorly—especially the material that teaches how to use the program and details its operational characteristics. The index is abysmal. I quickly compiled a list of important topics or concepts that don't appear in the index. For keywords that do appear, every mention of the word anywhere in the manual is indexed, with no highlighting to indicate where the main discussion of a command is, or where one might look for an example of a command's use. For a question on "logit" I found 71 different locations, including five separate references to table of contents pages!

The command syntax is a bit clumsy and hard to remember: even when I was testing the program fairly intensively I found that I had to look up many of the basics all over again if I stepped away from the program for just a few days. Here's an example of peculiar usage from the discussion of using temporary files: "In order to close a scratch file, give the `ROWS` command" (p. 94). Why should I remember `ROWS` to deal with scratch files? And neither "scratch files" nor "temp files" appear in the index to aid me when I do forget about `ROWS`.

RATS

For most of its life, Regression Analysis for Time Series (*RATS*) has been exclusively intended for time-series analysis, where it seems to be the clear favorite of most researchers. Version 3 and the newly released Version 4 (June 1992) introduced some cross-sectional tools, especially for panel data. Nonetheless, no one doing primarily cross-sectional work is likely to prefer *RATS* as their primary tool.

For time-series analysis, *RATS* has powerful programming capabilities and the broadest set of built-in procedures, at least for linear models.[7] Procedures are provided for ARIMA (Box-Jenkins) models, spectral analysis, Kalman filtering, and vector autoregressions. *RATS* also provides extensive and integrated recognition and handling of periodic data: yearly, quarterly, monthly, weekly, daily and even intraday (*SAS* provides even greater periodicity control). In addition, *RATS* is very popular for its flexible and effective time-series presentation graphics. Finally, *RATS* has an important advantage (shared by *Limdep*, *TSP* and *SST*): its primary authors over the years are themselves econometricians who use the program for empirical research.

[7]Version 4 introduced a generalized-method-of-moments procedure that offers much of the nonlinear capability *TSP* has provided for several years.

RATS has a number of drawbacks: awkward syntax (which is particularly difficult to use when programming extensions); very limited cross-sectional tools; limited tools for nonlinear models; and the absence of many fundamental data handling abilities.

Data Set Construction. I gave up without completing the data handling task in *RATS*. I figured out some tricks and believe that it could be accomplished, but no one using *RATS* for real work would ever bother. The program simply doesn't provide many of the standard tools for data set manipulation. For example, all data must be numeric; in my task I had to create externally a numeric identifier for each firm before reading the data into *RATS*. Similarly, I had 20 years of aggregate tax data to merge by year with my panel of firm data, but no facility for merging by an identifier variable ("match merging") is provided.[8] For cross-sectional or panel data handling, *RATS* is insufficient.

Even if your data set already exists in the form you want it, the *RATS* syntax for reading and writing is clumsy and hard to remember. To get a series requires two commands, first to open the file—and there are two different ways to do this for some files—and then to load the data. To save a new variable also requires two commands, first `store` and then `save`. This is frustrating for the occasional user.

Panel Data. My fixed and random effects estimations required about 15 lines of programming in the *RATS* command language. There is a complete example in the manual that can be used as a template, and the programming is facilitated by two helpful commands: `panel` and `pstats`. `Panel` accomplishes the fixed effects transformation in one step: it calculates within-group means and subtracts them from individual observations. For random effects `pstats` obtains the variance components and `panel` does the required quasi-differencing.

Even with these commands and the example in the manual, one needs a fairly good understanding of the details of the estimators to implement them. I also had to program to get the specification test statistics, and there is no example for this in the manual.

Simultaneous Equations. RATS is mostly good for a system of linear equations. The procedure handles seemingly unrelated regressions and three-stage least squares, and is especially good at handling cross-equation restrictions. Indeed, I recently used *RATS* for a paper because it was easy to specify various cross-equation restrictions. Robust standard errors are not available for the linear system estimator (`SUR`), but the new `NLSYSTEM` command in Version 4 (for nonlinear systems estimation) calculates robust standard errors, and can be used for linear problems.

The documentation for systems estimation is rather poor. The chapter on simultaneous equations has many examples but *not one* actually uses a

[8]One valuable feature appeared in Version 4 for the first time: the ability to create a permanent dataset that is a subset of existing data selected by the values of some variables, using `smpl (series =`X`)` where X is a selection variable of zeros and ones.

full-system method (three-stage least squares (3SLS) or seemingly unrelated regressions). To write down a 3SLS problem one must first slog through Chapter 9 ("Simultaneous Equations"), then each of three separate reference sections: `sur`, `equation`, and `instruments`.

Nonlinear Least Squares. A nonlinear least squares procedure is provided, with the ability to also estimate nonlinear models with instrumental variables. The syntax is clear and almost identical to that in *TSP* (which is the strongest program for nonlinear estimation).

RATS and *TSP* have been close competitors for years, with *RATS* stronger on time-series methods and testing, while *TSP* dominated nonlinear and maximum likelihood estimation. Recently the two have been converging. *RATS* Version 3.10 introduced a `maximize` command for general nonlinear (single-equation) models, including maximum likelihood. The Version 4 generalized-method-of-moments procedure adds a tremendous amount of power to *RATS* in an area that used to be one of its weakest points. (On the other hand, *TSP* now offers nearly all of the time-series tasks provided by *RATS*, as described below.)

Limited Dependent Variables. There are no procedures in *RATS* for hazard models, nor for duration models, event count data (like poisson), or censored and truncated samples (like Tobit). Of course, with the `maximize` procedure one can directly estimate the likelihood; this is what I did for my hazard model task. The manual has a discussion with examples showing how to estimate some limited dependent variable models.

Hypothesis Testing. *RATS* is very powerful for hypothesis testing, but (as with many tasks in *RATS*) the job is more difficult than necessary. For example, there are five commands for testing restrictions on estimated coefficient vectors; in *TSP* all of these tests can be performed with the single `analyz` command. Also, one specifies the coefficients that are restricted under an hypothesis by their position number in the regressor list (rather than by the regressor name) which is quite error prone (especially in a system of equations). Other test statistics can be calculated using the saved coefficient vectors and covariance matrices, the matrix algebra tools, and the provided procedures for evaluating critical points of standard distributions. The manual provides many programming examples for test statistics throughout, with one entire chapter devoted to the topic. The awkward syntax in *RATS* makes such tasks more trying than in, for example, *TSP*.

Ease of Learning and Documentation. *RATS* is rather difficult to learn, primarily because of its difficult syntax. Also, the writing in the manual is uneven, often presented in disjointed, poorly organized subsections. However —prior users take note—Version 4 is a major improvement. The index used to be dreadful; it is now quite useful. Cross-references between related topics have been added. And the programming language has been simplified in a couple of places.

The manual now has a tutorial chapter that works through each of the steps of a complete econometric project. Nine chapters offer more extensive tutorials on topics such as various regression models, hypothesis testing, vector autoregressions, panel data, and programming. The manual is filled with command examples and several discussions of econometric theory and the assumptions for different procedures. If the task you need to perform appears in an example—and frequently it will—then it is relatively straightforward to use *RATS* on an occasional basis. If you need to do something that isn't illustrated, it may take considerable effort.

SAS

SAS is unique because of its sheer size and scope; *SAS* is almost an industry unto itself. It runs on more platforms than any other program, and has a much larger user community. However, most of those users are not econometricians, but other social and physical scientists.

If this review were based on comparing the length of the features list, *SAS* would easily dominate. But in trying to be all things to all people, *SAS* stumbles in several features important to economists. For example, much of the (voluminous) documentation is written in a jargon that is drawn from the analysis of variance tradition more familiar to pyschologists, sociologists and others. Data are always treated one observation at a time, so there is no syntax for referring to lagged values; you must create a new variable. *SAS* tends to be further behind current estimation and testing methods than some of the other programs, and relatively weak on limited dependent variable estimation.[9]

The PC version of *SAS* is slower than the other programs reviewed here (except *Limdep*) and makes inefficient use of extended memory (it ignores any extra memory after the first two megabytes). And it is a disk hog: the four econometrics-related modules I used to tackle the problems in this review require 16.5 megabytes of storage.

Though I give time series methods short shrift, I should mention that *SAS* compares favorably with *RATS* and *TSP* in its capabilities. *SAS* can handle all sorts of data periodicities, and provides some powerful tools for massaging periodic data. There are statistical tools for ARIMA, vector autoregressions, and spectral analysis. However, although I cannot provide a user's viewpoint, I have noted that *SAS* is rarely used by time-series econometricians, at least in academic settings.

Data Set Construction. *SAS* is hard to beat for data handling. Some tasks are easier in *SST* and *Stata*, and some time-series transformations are easier in

[9]To be fair, *SAS* is under continuous major development. In response to my initial draft, the SAS Institute replied with evidence that they are close to plugging many of the gaps I've identified. I limited this review to already available procedures, but *SAS* users are well-advised to keep up with new developments. Unfortunately, *SAS* is not effective at keeping its users informed, particularly at site-licensed institutions. I have been a regular user of *SAS* for a decade, and was startled to learn how many new econometric procedures had been developed of which I had not been made aware. Most of the other vendors are much better at communicating new developments.

RATS and *TSP*, but the others simply don't have the flexibility and depth of *SAS*.

One feature of *SAS* is very important: because it works with just one observation at a time, the only limit on dataset size is disk (or tape) storage, not memory. (The converse is that working with the entire data set in memory is what allows the other programs to perform some data tasks with greater aplomb.) This capability to look at data one observation at a time is crucial if you have a six megabyte data set to boil down on a machine with two megs of random access memory. In fact, most econometricians I know use *SAS* almost exclusively for data processing before doing their analyses in another program, especially if they have large data sets. Also, many data providers are now offering their data sets in *SAS*'s format: it is much easier and less error prone to access and extract data in *SAS* than in, say, FORTRAN.

Panel Data. When I tackled the panel estimation task with *SAS*, I was surprised to find no built-in routine for random effects estimation. This despite *SAS* having an extraordinarily complete capability for the closely-related analysis of variance (ANOVA) methods common in psychology, sociology and other fields. To estimate the random effects model I had to go through several steps: run ordinary least squares, estimate the variance components, quasi-difference the data using the variance components, and then finally obtain the estimates from a regression on the transformed data. One bright spot: there is a procedure to estimate the variance components, which is the most likely place for a programming or conceptual error.

SAS handled the fixed effects model with no problem, but pity the poor user who has to *find* the procedure for the first time. Rather than a separate procedure or option in the regression procedure, one must wade through the ANOVA jargon of the `proc glm` (generalized linear model) documentation and figure out that the `absorb` option does the trick. (If this sentence was incomprehensible to you, that's the point!) And then, of course, since there is no random effects estimator in *SAS*, there is no built-in procedure to perform a Hausman specification test comparing the two (Judge et al., 1985, p. 537).

Simultaneous Equations. I was able to estimate my simultaneous equations problem using `proc syslin` in the ETS module. However, there is no provision for computing standard errors that are robust to either heteroskedasticity or serial correlation of unknown form. *SAS* does offer a wider variety of linear systems estimation methods than do the other programs (for example, k-class), and reports one test of overidentifying restrictions.

Nonlinear Least Squares. My NLLS task provides an excellent illustration of *SAS*'s strengths and weaknesses for the "serious but occasional" econometrician. There is a `proc nlin` in the Stat module for nonlinear least squares. Its documentation illustrates the bulkiness of the manuals: 35 pages for the one command. For this price you get seven detailed examples with output, and a thorough discussion of the five "available" algorithms and possible numerical problems. The derivatives of the nonlinear function must be provided by the user (as in *SST*, but unlike *TSP*). The output is limited: one can obtain some regression diagnostics (studentized residuals and leverages for diagnosing outliers and influential observations; see Belsley, Kuh and Welsch, 1980) but no test statistics for serial correlation or heteroskedasticity. There is no option for computing robust standard errors.

However, there is another procedure altogether—`proc model` in the ETS module, documented in a separate manual—that *also* does nonlinear least squares (as well as nonlinear systems estimation and solving)! `Proc model` is a powerful procedure, with features quite different from `proc nlin`. Notably, the derivatives do not have to be provided by the user; in fact, `model` is the only program besides *TSP* that itself does analytical derivatives for estimation (but not for testing). This procedure will report a Durbin-Watson statistic (though still no heteroskedasticity statistics, and no corrections for robust standard errors). This is typical of *SAS*: remarkable breadth in the available tools, but also much redundancy, and substantial difficulty in finding the right tools. It would certainly help to have a master index by topic that cross-referenced all of the many manuals.

Limited Dependent Variables. *SAS* has a procedure for the Cox model, though it must be specially requested and is not documented in the manuals for the current PC version (6.04). Despite having a procedure for this sophisticated model *SAS* is rather weak overall on limited dependent variable models, with no procedures for several models commonly used by economists. What is further surprising, given the scope of *SAS*, is that it offers no procedure for performing maximum likelihood estimation on a user-defined likelihood.[10] (*SAS* does offer a very capable matrix programming language with which adventurous users could program an MLE routine.)

Hypothesis Testing. *SAS* was very capable for my tests of linear restrictions on parameters in a single equation. With `proc syslin` one can also test cross-equation restrictions in a system, and estimate with equality restrictions imposed. There is no provision for testing *nonlinear* hypotheses, though one could program them with the matrix algebra module.

Unfortunately, *SAS* is rather sparse on other tests. No heteroskedasticity tests are provided (see Judge et al. 1985, pp. 445–454); the Durbin-Watson statistic is available but not the alternatives for use when Durbin-Watson is inappropriate. Specification tests are not directly implemented.

Ease of Learning and Documentation. The vast number of commands and bulky manuals convince many people that *SAS* is hard to learn. I disagree for some tasks. In particular, learning data handling in *SAS* is straightforward. Most functions are obvious from command names making the table of contents an excellent starting point. The documentation is extraordinarily detailed and

[10]The SAS Institute informed me that a nonlinear programming procedure in the OR (operations research) module could do this, but I did not include OR in this review because it is not designed for econometrics; I've never seen it used by an econometrician, in fact.

filled with examples. I have had research assistants working productively in *SAS* by the end of a day. The manuals, however, are sold separately at steep prices. And there is no introductory tutorial example that takes the user through all of the steps for an econometric project.

Things get much harder upon moving to econometric analyses. Here the variety of procedures is bewildering: at least nine different procedures can estimate a linear regression. For my panel data task, it took quite an effort to convince myself that there *wasn't* a direct implementation of the random effects model I wanted–I kept thinking if I looked at just one more manual I would find it hidden in some obscure procedure.

SST

SST is the product of two practicing research econometricians. Their program is "lean and mean," with fewer total commands and thinner manuals than any of the others. There are a few major gaps in econometric methods, although the programming language is powerful enough that many of the gaps can be plugged if desired.

Three features make *SST* unusually powerful despite its limited range of procedures. First, every procedure can be run easily on a subsample based on observation numbers or a logical condition (for example, if $x > 0$). Second, it offers a maximum likelihood procedure that accepts a user-defined likelihood function, permitting the user to readily estimate a wide variety of models that are not pre-programmed. Third, the program has a rather powerful matrix programming language that one can use to construct estimators and test statistics that are not preprogrammed. *SST* does not provide as many matrix manipulating functions as *Gauss*, but it has enough power and flexibility for most statistical operations.

Data Set Construction. *SST* is a pleasant surprise on data handling. *SST* is easier than *SAS* and sufficiently powerful for many standard tasks. *SST* has a clean syntax for data handling with emphasis on the tasks that econometricians regularly face.

SST permits lagged values with a natural syntax (as do *Limdep*, *Stata*, *TSP*, and *RATS*, but not *SAS*). Sorting and merging is fast and easy (these tools are more powerful in *SAS*, but not available at all in *RATS* or *TSP*). *SST* quickly extracts summary statistics by subset of the data, a feature not fully available in *RATS* or *TSP*. For example, I wanted to drop all observations for which I did not have a full 20 years of data; with one command I counted the number of observations by firm, and with one more I dropped the culprits. A similar procedure was more complex in *SAS*. *SST* handled each step in my data task directly, closely mimicking the way I conceived the problem. Thus, the programming required was minimal and straightforward.

Panel Data. There are no direct implementations of fixed or random effects models in *SST*. The `aggregate` command collected means by firm, allowing me to accomplish the fixed effects estimation—but I had to know the formula for the estimator. Piecing together random effects estimators by hand is more difficult because (as in *SAS* and *RATS*) one must first estimate the variance components from the residuals of a preliminary regression, then pseudo-difference the data using a variance components weight, and finally estimate.

Simultaneous Equations. *SST* cannot directly estimate a system of equations. A general purpose instrumental variables estimator is available for single-equation estimation (including two-stage least squares). One could implement three-stage least squares by stacking the model and transforming the data in a two-step procedure that uses only single-equation methods, or program a three-stage least squares estimator directly, but either involves a lot of work.

My simultaneous equations task called for robust standard errors. *SST* does implement heteroskedastic-consistent standard error calculations in some of its pre-programmed procedures, but not for user-defined maximum likelihood estimation. There is no provision for standard errors robust to autocorrelation of unknown form.

Nonlinear Least Squares and Limited Dependent Variables. *SST* provides several limited dependent variable estimators, but neither nonlinear least squares nor a Cox proportional hazards procedure (though it does implement the less general exponential regression model). I was able to program both as maximum likelihood problems. Since NLLS is a pseudo-likelihood problem I had to know a trick or two to use MLE; the manual does not help with an example of NLLS for guidance.

Hypothesis Testing. *SST* does not have the convenience of a post-estimation testing command for hypotheses on the coefficients of a model. The general tools needed to test such hypotheses are provided and easy to use, if you know the matrix algebra formula for your test statistic. In particular, it is possible to store the variance-covariance matrix for the estimated coefficients of any procedure, and the matrix manipulation language allows easy calculation of most any test statistic. Still, most testing is more convenient and less error prone in the other programs.

Ease of Learning and Documentation. I was able to get programs running quickly under *SST*. The first 70 pages of the *User's Guide* are a good tutorial for the novice. The manual is liberally sprinkled with examples, though they tend to be one-line illustrations of the command just defined. It would be helpful to have a separate tutorial that ran through a complete data manipulation and analysis exercise.

The main stumbling block to learning and using the program is the incompleteness of the current manual, which is a work-in-progress. My copy was provided by the authors in late 1991, with a 1988 copyright. Despite this three-year lag, it has numerous "not-yet-written" placeholders. It is difficult to find things: the table of contents for the *User's Guide* is quite thorough, but neither of the two manuals offers a topical index. The division of material between the manuals is odd, too: the *Reference Manual* documents commands and options, but the reference material for functions, programming constructs and built-in variables all appears at the end of the *User's Guide*. The on-line help is very complete—indeed, it appears to be a complete copy of the reference manual. However, numerous errors and confusions appear throughout.[11] If the manuals were completed and improved, with more thorough examples and careful indexing, then *SST* would be one of the easiest programs to learn and use.

Stata

Stata is a lesser-known econometrics program, but its users praise it enthusiastically. Like *SAS*, *Stata* serves a large non-economist user community, but it is more congenial to economists. Four powerful data options can be attached to any of the statistical commands, enabling simple and understandable programs. These are: `by`, `if`, in, and `=`. Any procedure can be run separately on subsets of the data identified with a `by` variable, an `if` logical expression, or observation numbers `in` a range.[12] The `=` option allows the assignment of weights to the data before performing the procedure.

Stata is one of the easiest to learn powerful statistics programs, with especially good graphics capabilities. But among the programs reviewed here it is the least sophisticated in terms of modern econometric testing procedures. It is also the most aggressively cross-sectional.

Data Set Construction. Like *SST*, *Stata* was a surprise and a delight on the data manipulation task. *Stata* is very similar to *SST* in data capabilities; indeed, essentially all of the specifics mentioned above for *SST* apply to *Stata* so I will not repeat them here.

In a couple of ways *Stata* is less friendly. Many of its data handling commands can't operate on more than one variable at a time. To recode missing values I had to either repeat the command for each variable or program a little procedure; this task took one built-in command in *SST*. More important, *Stata* doesn't give consistent handling of missing values, which can cause serious problems. In some situations an operation on a missing observation creates a missing value—the expected result. In other places—particularly in aggregating or summarizing variables, missing values are treated as zeros, with no warning to the user. On the other hand, *Stata* offers more data

[11]For example, help on `read` (for entering raw data) has a subtopic for the `if` option, which suggests that a data subset can be created according to logical selection conditions as they are read (this is a handy memory and time-saving trick available in *Stata* and *SAS*). In fact, `read` ignores the `if` option.

[12]In *SAS* one has to run a separate "data step" to create a temporary data set in order to subset by the latter two criteria; this extra work does have the advantage of making the output log very clear on the data subset currently under analysis.

handling features than *SST* and provides superb graphical tools for both exploration of data and presentation of results.

Panel Data. *Stata* has no provisions for standard panel data estimation techniques. Thus, to accomplish my task, I had to program the estimators directly. I am surprised these methods aren't standard since *Stata* is widely used by labor economists and demographers. As consolation, one can write a routine as a reusable command that operates just like a built-in command. Thus, the commands I wrote for fixed and random effects became permanent extensions to the command set, and I can call them like built-in commands if I need them again (*Stata* finds them automatically). Such commands can require nontrivial programming, of course.

Simultaneous Equations. *Stata* has no built-in procedures for estimating systems of equations of any sort (linear or nonlinear, three-stage least squares (3SLS), seemingly unrelated regressions, or any other generalized method of moments estimator). The only nod towards simultaneity is a single-equation instrumental variables estimator. It is possible, in principle, to program a 3SLS estimator in *Stata*, but because there are no tools provided for Cholesky factorizations or matrix multiplications, coding the generalized least squares transformation that turns a system of 2SLS estimates into 3SLS would be very difficult. Programming test statistics would be difficult, as well. (Heteroskedasticity-consistent standard errors are obviously not available for simultaneous equations, but they are available for linear regressions, logits and probits, though they are hidden in a procedure named `huber`—a name not associated with these methods by most economists.)

Nonlinear Least Squares. *Stata* has no built-in procedures for nonlinear regression models, but a supplemental procedure is available. *Stata* has a very active user community with a bimonthly newsletter full of user-written procedures, examples for teaching, reviews, etc. (*Gauss*, *RATS*, and *SAS* also have periodicals.) Most of the material is aimed at noneconomists, but some gems appear such as a good nonlinear least squares procedure in the May 1992 issue. (The source code for user-contributed programs in the newsletter is available on disk for an additional $70 per year, which must be more than ten times marginal cost.)

Limited Dependent Variables. *Stata* is quite strong on hazard rate and survival function problems. The built-in Cox proportional hazard procedure can handle observations that are censored at either end of the sample period, or both. There are also procedures to estimate and graph the baseline hazard and survival rates from the Cox model, as well as various nonparametric estimates of survival curves.

Stata does not provide tools for directly estimating duration models (or the closely related event count models, such as the poisson). *SST* and *Gauss* provide procedures for duration models but not for hazard models.

Hypothesis Testing. *Stata* is rather weak on testing. It has a `test` command, which tests linear hypotheses using the estimated information matrix from the most recent regression procedure (linear regression, Cox model, logit or probit). Any general linear hypotheses can be tested, singly or jointly, but nonlinear hypotheses are not allowed. The manual incorrectly implies such tests are not possible ("a limitation caused by the statistics of the problem, not by a limitation of *Stata* itself," vol. 1, p. 273); in truth asymptotic testing of nonlinear hypotheses is straightforward but *Stata* chooses not to support it.

There are even more serious limitations on hypothesis testing in *Stata*. The program makes no provisions for matrix algebra, and it is not possible to capture the variance-covariance matrix as a variable after a regression. Thus, it is hard to generate any test statistics *other* than the built-in tests of linear hypotheses. Further, very few diagnostic tests are available in *Stata*. For example, there are no tests for heteroskedasticity. A Hausman (1978) specification test is not provided for the instrumental variables procedure.

Ease of Learning and Documentation. I initially found *Stata* frustrating to learn, but most of my complaints were fixed by the Version 3 manuals. The manual now has a detailed sample session, and a lengthy section devoted to tutorials on over a dozen general topics (like using categorical variables; getting data into *Stata*; estimation/prediction/testing). There is also an excellent set of on-line tutorials, and superb on-line help complete with examples.

Once learned, I found *Stata* very friendly and easy to use. The documentation is clearly written, and every feature of a command is illustrated with an example. The documentation provides quite a bit of explanatory material on statistical issues. For example, after describing how to convert a string categorical variable into a numeric variable, and then into a set of dummies, there is a four-page discussion on estimating a regression with dummies, doing joint F-tests, and interpreting the dummies; this includes a useful discussion relating analysis-of-variance (ANOVA) to regression with dummies. The abundance of examples and statistical digressions makes the manual difficult to use as a *quick* reference, but the extra material is probably worth it for the occasional user.

Stata has several idiosyncrasies, good and bad. For example, there is a very complete set of data types (byte, integer, float, double, etc.) which is good for conserving memory but can lead to confusion and errors. It is easy to make errors in logical comparisons with data stored in single precision because calculations are done in double-precision. I got incorrect results when I tested for `x == 0.0001` (a crucial error because this was the missing value code in my Compustat data). I had to learn to test `x = float(0.0001)` instead, which is hardly obvious.

TSP

TSP has long been a model for a powerful program with a simple, easy syntax; indeed, a number of other programs not reviewed here have baldly imitated it. *TSP* has also long dominated the competition for nonlinear estimation. It has automatically handled analytical differentiation to generate gradients and nonlinear hypothesis tests since the 1960s, yet none of the other programs reviewed do that even today (except *SAS*, which uses analytical derivatives in its `proc model` for nonlinear systems, though not for hypothesis testing nor in `proc nonlin` for estimating single nonlinear equations).

Despite its original name (Time Series Processor) *TSP* has always contained a mix of cross-sectional and time-series methods, probably reflecting the fact that its original authors are practicing research economists, one a microeconomist and the other a macroeconomist. This has been something of a problem for the program's popularity. Most users tend to specialize, and unless a user needed nonlinear methods, other alternatives usually offered more complete tools for either cross-sectional or time-series analysis. With the most recent revisions, however, *TSP* has developed into a program with tremendous depth as well as breadth, and it warrants a reconsideration by many researchers. In particular, Version 4.2 offers most of the time-series capabilities of *RATS* (ARIMA, VAR, and Kalman filter, but not spectral analysis), and often in an easier-to-use style.

Data Set Construction. *TSP* is better than *RATS* for most data handling problems, but not much. As the authors themselves say in the *User's Guide*, "it is possible to do fairly complex data transformations in *TSP*, but you should keep in mind that there are probably many other more efficient languages available for this" (Ver. 4.2, p. 113). I was able to complete my data task with *TSP*, but it required some strange tricks that should be avoided in ordinary work. The main difference between *TSP* and *RATS* in data handling is that the former is not as closely wedded to the dated time-series view of data. So, for example, panel data sets with a different number of observations per unit (say, household) are allowed. But as with *RATS* there is no provision for merging by an identifying variable, and character data are not allowed.

Panel Data. *TSP* has the best panel estimation procedures of any of the programs, except perhaps *Limdep*. One command generates the full set of standard estimates: ordinary least squares, between, fixed effects and random effects. A good set of test statistics are also reported. This was more than adequate for my task.[13] In addition, *TSP* is the only program that provides panel data estimation when there is serial correlation (again, except *Limdep*). The manual even presents a careful description and thorough example of how to construct Chamberlain's (1982) robust estimator without the assumptions of conditional homoskedasticity or independence over time, a leading-edge method still rarely used in practice.

[13] I did run into one problem: the estimated variance components were calculated with the wrong degrees of freedom. This raises another point in *TSP*'s favor, however: the main programmer is available by phone or electronic mail, and he is *extremely* responsive to user queries for help. In this case he immediately acknowledged the problem, recompiled the corrected program and mailed a new disk to me. I found similar responsiveness from the *Stata* and *Limdep* authors, and a good response from *RATS*, *SAS*, and *SST*. Technical support from *Gauss* is very uneven.

Simultaneous Equations. *TSP* is the dominant program for simultaneous equations estimation (a few additional but rarely used methods are available in *SAS*). With just the `lsq` command the user can get ordinary least squares, 2SLS, 3SLS, and seemingly unrelated regressions estimates for both linear and nonlinear models. It was easy to perform my test problem, including the calculation of robust standard errors; *TSP* was the *only* program that offers robust standard errors throughout as a standard feature. The manual contains an extensive and clear discussion of the objective functions and statistical properties for each of the several models.

A procedure for generalized method-of-moments (GMM) estimation was recently added, which extends systems estimation to include an option for robustness to time-dependence as well as conditional heteroskedasticity.

Nonlinear Least Squares. The heart and soul of *TSP* is its power for handling nonlinear estimation. The two key features of *TSP*'s nonlinear capabilities are: (1) the standard estimation commands are the same for linear and nonlinear models, making it much easier to learn and use; and (2) *TSP* automatically does analytical differentiation.

The importance of built-in analytical differentiation is hard to overemphasize for nonlinear estimation. Compared to programs that use numerical derivatives (*Limdep*, *RATS*, optionally *Gauss*) analytical derivatives are more than twice as fast, and evaluating derivatives is the time-hog for nonlinear estimation. Moreover, programming errors can be avoided. In *SST* and *SAS*'s `proc nlin` (and optionally *Gauss*) the user has to correctly write the (often very many) derivatives down on paper, and then correctly program the expressions. When I've used one of these a large fraction of the time spent on nonlinear estimation is devoted to debugging derivatives.

Limited Dependent Variables. *TSP* does not implement the Cox model, but I could estimate it with the maximum likelihood procedure. The *TSP User's Guide* contains 12 detailed examples of maximum likelihood models, providing ample guidance on coding functions of various types. The manual also coaches the user on tricks for writing the likelihood in a way that minimizes computation time.

Hypothesis Testing. I think *TSP* is the clear winner on built-in hypothesis testing procedures. With one command it can generate the longest list of diagnostics for linear ordinary least squares of any of the programs, including the usual t and F tests, five tests for AR1 serial correlation, two tests for higher order autocorrelation, the Dickey-Fuller test for a unit root in the residuals, a test for ARCH(1) residuals, a Chow test for a structural break in the model (with user-specified break point), and three tests of heteroskedasticity. *TSP* also shines when testing hypothesized restrictions on the parameter values. One command (`analyz`) can be used to test any general linear or nonlinear hypothesis restricting the coefficients of a model. None of the other programs will directly test nonlinear hypotheses because of the need to calculate the derivatives of the restrictions (again, except *Limdep*, which however uses numerical derivatives for the calculation).

TSP's matrix programming language can be used to calculate test statistics not provided directly. The syntax is almost as natural as *Gauss*'s, and the functions provided for matrix algebra are almost as extensive as those in *SST* (but nowhere near those in *Gauss*).

Ease of Learning and Documentation. Only *Stata* is somewhat easier to learn than *TSP*, but it is also much more limited in power. The *TSP* manuals are first-rate. The tutorial chapter in the *User's Guide* is excellent: clearly written, filled with precisely the information needed to get started, and closed with an extended example that takes the reader through a complete, realistic econometric project from data to estimation to testing. The rest of the guide is filled with useful explanations of econometric and numerical methods, detailed and realistic examples, and discussions of programming technique, yet it is concise and very readable. This manual would make a good companion text to a first-year graduate course in applied econometrics.

Gauss

Gauss is the least similar of the programs reviewed. I include it because it is a very successful program to which large numbers of advanced econometricians are addicted, and it is often recommended to graduate students. *Gauss* is superb if you routinely need non-standard estimators or test statistics. It is also useful for nonstatistical numerical analyses, such as solving nonlinear systems of first-order conditions, plotting numerical comparative statics, or solving differential equations.

The other programs emphasize pre-programmed procedures and then provide some programming capabilities to extend the intrinsic command set. *Gauss* is instead a high-level programming language, supplemented by some pre-programmed procedures. Using *Gauss* is like using C or FORTRAN to write statistical analysis programs, with one crucial difference: *Gauss* is a *matrix* programming language: every variable is a matrix, and an extremely complete set of matrix algebra tools is available, with an emphasis on those needed for statistics.

The vendor sells optional applications modules that implement many standard procedures. However, the modules are costly and suffer from more frequent errors or "bugs" than do the other programs reviewed here. The applications are harder to use than the built-in commands of other programs, and technical support is spotty.[14] They are reasonably broad (comparable to *Stata* and *SST*), but not deep enough. There are too few test and descriptive statistics, and too few options for diagnosing and dealing with disturbance terms that are not independent or identically distributed.

[14]For example, I hit a bug in one of my review tasks: one variable's coefficient had a t-ratio of 125, but *Gauss* reported the corresponding tail probability—which should of course be almost exactly zero, as 2.0! I reported this bug but never received a reply. However, the bug was fixed in Version 3.

Gauss performed less well on almost every one of my tasks than did each of the other programs. There were a few bright points, however. And please remember: when I say *Gauss* can't do something or is limited, that means there is no built-in command for the task. More than any of the other programs, *Gauss* can be extended to perform virtually any econometric task.

Data handling has long been criticized by users. The simplest tasks required nontrivial programming, with the side effect of creating many opportunities for coding errors. Version 3 is a substantial improvement because it includes a `dataloop` mechanism that is similar to—though much more limited than—the easy-to-use data step toolkit in *SAS*. *Gauss* is now more benign than *RATS* or *TSP* for data work that requires merging, sorting and transforming, but still falls far short of *SAS*, *Stata*, and *SST*.

Simultaneous equations estimation is a good example of what's wrong in the *Gauss* application modules. The Durbin-Watson statistic, R^2, and the coefficients and their standard errors are reported, but that's about it. No χ^2 statistics to test the overidentifying restrictions; no reporting or diagnostics on the estimated cross-equation variance-covariance matrix used for weighting in the 3SLS estimation. It is not possible to input a weighting matrix externally, which is necessary for a number of standard tests. There are no tests or corrections for heteroskedasticity, and no adjustment to produce heteroskedastic-robust standard errors.

One highlight is the panel data module, which provides each of the standard estimators used in the economics literature (TSP and Limdep are the only other programs that provide all of these directly). A good set of test statistics are generated for the maintained hypotheses that differ across these models.

Gauss is difficult to learn for occasional econometric applications. The manuals are oriented toward programmers; for example, the tutorial deals only with programming and there is no tutorial for the applications modules. Examples are minimal and poorly commented (except in the new manual for the Linear Regression module).

Choices

What program would I take to a desert island? Probably *Gauss* because in the end I could handle any problem (including non-econometric numerical analyses), but I enjoy programming as a hobby. I'm tempted to say *TSP* except that I would need some other tool for data handling. If I could take two programs they would be *SAS* and *TSP*. I would expect a primarily time-series econometrician to be torn between *RATS* and *TSP*. *SAS* is an alternative to *Gauss* as a single comprehensive toolkit, but it is probably too overwhelming for the *occasional* user.

What do I use now? Mostly *SAS* for data handling, and *Gauss* for everything else, although I have also used *TSP* and *RATS* for recent projects, and am currently delighted with *Stata* for a project with lots of data handling and general equilibrium computations (but little estimation). Although I rely heavily on *Gauss*, I have moved most of my work to a UNIX workstation and will wean myself from *Gauss* towards *TSP* unless *Gauss* introduces a multi-platform workstation version very soon.

What would I recommend to an economist who is about to do a first and last econometric project? *Stata* or *SST* if they have the right specific procedures for your project, or *TSP* if your data set is already in good shape. For the serious but occasional econometrician, it is hard to narrow a recommendation further. The best tool will depend on the types of analysis you need (especially whether you use a limited number of standard methods, or need the greater breadth of *Limdep*, *TSP* or *RATS*), how much data manipulation you do, and how much you like programming.

■ *I am grateful to David Lam, Jim Levinsohn, Matthew Shapiro, and Gary Solon for helpful comments and advice. Timothy Taylor provided wonderful editorial assistance. I sent a prior draft of the review to each of the software vendors to check for accuracy; any remaining errors are my own.*

References

Amemiya, T., *Advanced Econometrics.* Cambridge: Harvard University Press, 1985.

Belsley, D. A., E. Kuh, and R. E. Welsch, *Regression Diagnostics: Identifying Influential Data and Sources of Collinearity.* New York: John Wiley, 1980.

Chamberlain, G., "Multivariate Regression Models for Panel Data," *Journal of Econometrics,* January 1982, *18*:1, 5–46.

Hausman, J. A., "Specification Tests in Econometrics," *Econometrica,* November 1978, *46*:6, 1251–1272.

Judge, G. G., W. E. Griffith, R. C. Hill, H. Lütkepohl, and T.-C. Lee, *The Theory and Practice of Econometrics.* New York: John Wiley, 1985.

Maddala, G. S., *Limited-Dependent and Qualitative Variables in Econometrics.* Cambridge, UK: Cambridge University Press, 1983.