

A review of Stata 8.1 and its time series capabilities

Christopher F Baum*

October 16, 2003

1 Introduction

Stata is a general-purpose statistical package with an accessible programming language. Its latest major release, 8.0, became available in January 2003, and represented the most thorough redesign of the program's capabilities since its inception. This redevelopment has provided a number of features that go beyond version 7.0, as reviewed by Kolenikov (2001). Other changes in the Stata product line were made during the life of version 7.0, as described below. Since upward compatibility is a high priority for Stata's developers, the vast majority of the features described in that review are accessible in the current version as well.

Two major changes appeared in version 8.0: the accessibility of all commands from menus in the graphical user interface, and a complete overhaul of Stata's graphics capabilities, involving the development of an object-oriented graphics language which allows for the extension and customization of graphics. In July 2003, the first minor release of Stata, version 8.1, became freely available to purchasers of version 8.0. The 8.1 release formally adds support for "plug-ins," or routines in compiled C code in addition to Stata's own interpreted language. Users are able to write their own "plug-ins" to perform computationally burdensome tasks for which interpreted code is excessively slow.

*Department of Economics, Boston College; Academic Technology Services, Boston College; Associate Editor, *Stata Journal*. I am grateful to Bruce McCullough for editorial comments, Vince Wiggins for constructive suggestions on the use of new code features, Nicholas J. Cox for assistance with the graphics code, and Paula N. Arnold for her review of this document. Remaining errors are my own.

2 Overview of Stata 8.1

Stata is produced by the Stata Corporation (<http://www.stata.com>) of College Station, TX (800-STATA-PC). It is offered in three versions: (1) a student version, limited in the number of variables (99) and observations (1,000), but otherwise complete in functionality; (2) Intercooled Stata, the standard version; and (3) Stata/SE (Special Edition). The latter version debuted during the life of version 7.0 in response to users' needs to analyze larger data sets. Stata/SE allows significantly more variables in a data set (32,767) and supports larger matrices (up to 11,000 rows or columns). In contrast, Intercooled Stata supports up to 2,047 variables in a data set, with the number of observations limited by available RAM (technically, to 2.147 billion), as the entire data set is held in memory. The Intercooled version allows matrices of up to 800 rows or columns. The limits on matrix dimension, even in Intercooled Stata, are not particularly onerous, as matrix-language commands are provided to compute, for instance, cross-product matrices from any number of observations (i.e. one may form $X'X$ even when the variables to be considered as columns of X have more than 800 observations).

Stata is available for a wide variety of platforms and operating systems, and is aggressively cross-platform; only a single, slender *Getting Started* manual is OS-specific, with all other documentation being platform-independent. Stata may be used on Windows (95 or better), Macintosh OS X, Linux on Intel, and many flavors of Unix, including those with 64-bit processors. Student versions are sold with one-year licenses, where all other versions are sold with perpetual licenses. Upgrades within the life of a version are free, as is technical support. Current prices range between \$965-\$1,095 for commercial users (\$1,215-\$1,345 for Stata/SE), or \$469-\$599 for single-copy academic users (\$615-\$745 for Stata/SE). Substantial discounts are offered for volume purchases, network licenses and educational lab environments. Much more attractive single-copy prices are available to academic users in institutions who have established "Grad Plans" for student and faculty purchase. A Base Documentation Set is bundled with all but "Grad Plan" copies, for which it is available separately. The set consists of the four-volume *Base Reference Manual*, *User's Guide* and *Graphics Reference Manual*. The higher prices above bundle the Full Documentation Set, and most importantly for effective use of Stata, include the *Programming Reference Manual*. As an innovation in version 8, the four-volume Reference Manual no longer contains a description of every Stata command, but focuses on commands of general interest. Five new manuals describe sets of commands in specialized areas: *Survey Data*, *Time-Series*, *Cross-Sectional Time-Series*, *Survival Analysis and Epidemiological Tables*, and *Cluster Analysis*. These manuals, included in the Full Documentation Set, may be purchased separately by those users interested in these methodologies.

2.1 Evaluation by Berk's criteria

Berk (1987) suggests a long list of criteria by which a statistical package might be evaluated. How does the current version of Stata stack up on these criteria?

- Documentation: the manual is fully descriptive, with mathematical detail of the algorithms used and examples of each command. All datasets referenced in the manual are freely available over the web. The full definitions of each command, with brief examples and hypertext links to related commands, are available via on-line help.
- Language: the package is fundamentally based on a command-line environment, and the command syntax is coherent and common across commands.¹ In its GUI version, the package has an internal editor; alternatively, any external text editor may be used in the interactive environment. When using the GUI's menu system to generate commands, the syntax of the commands may be reviewed. A sequence of interactive commands may be saved as a text file.
- Data handling: the package is very efficient at reading and writing in its own binary data set format, which is usable across platforms. Facilities for ASCII data input are thorough; data dictionaries may be used to define complex data set formats. Unlike some other packages, Stata has a full array of byte, integer, floating-point and character datatypes, permitting data to be stored in the most economical format (which it automatically determines). String variables may be used, and labels may be defined for variables and values. Random data may be readily generated for a number of probability distributions.
- Data manipulation: datasets of any size may be examined and modified with a built-in data editor in the GUI. The package can easily generate sets of dummy variables and provides a comprehensive set of data-handling functions for both numeric and string variables with sophisticated parsing facilities for the latter. A very powerful set of facilities is provided for the handling of panel / longitudinal data and working with data defined on a time-series calendar for a number of frequencies.
- Graphics: the package produces publication-quality graphics in a number of "schemes" corresponding to common publication formats. Graphics are highly customizable, and plots of various sorts may be superimposed and juxtaposed. Graphics files are produced in a native format and may be translated within the program into Portable Network Graphics,

¹Stata's web site indicates that the software's accessibility complies with the requirements of Section 508 of the Rehabilitation Act of 1998.

PostScript, Encapsulated PostScript, Windows Metafile, Windows Enhanced Metafile, PDF, and PICT, depending on platform.

- **Procedures:** Stata provides a full-fledged high-level programming language in which user-written procedures can be developed and made as easily accessible as built-in commands.
- **Output:** the standard output may be produced as ASCII text or as styled text in Stata's own SMCL markup format. A low-level `file` command may be used to generate customized text or binary output.
- **Customizing:** a very large fraction of the package's source code, in its own language, is accessible to the user and may be copied and modified for customization. The package has a reasonably powerful matrix language with no support for complex arithmetic.
- **Support:** telephone and email support is free to registered users; as all but student licenses are perpetual licenses with no annual maintenance fees, the support is not time-limited. This compares most favorably with other packages that require an annual payment to receive support. New versions of Stata appear every couple of years. Users are encouraged to purchase these upgrades, but are free to continue using the original version which they have licensed. Users with complex problems often receive direct support from developers who are highly trained in their subject field (e.g. developers with doctorates in econometrics or biostatistics).
- **Overview:** the package is appropriate for the vendor's targeted audiences: "medical researchers, biostatisticians, epidemiologists, economists, sociologists, political scientists, geographers, psychologists, social scientists and other research professionals needing to analyze data." The emphasis in the package's documentation and fora—user group meetings, journal, website, etc.—is clearly placed on the research professional and not on the business user looking to draw 3-D bar charts of sales by region. Stata's language is readily learned, perhaps more easily than other packages' syntax, and it provides menu-driven access to all commands in the GUI. The simplicity of its use, coupled with its high-end programming capabilities, makes it an excellent tool for instruction of statistics and research methods at any level. The advanced student who learns its use will be able to carry out doctoral-level research efficiently and be able to generate a fully reproducible analysis of the data.

3 Capabilities of the Stata environment

Stata advertises its strengths in three areas: data manipulation, statistics, and graphics. It provides excellent facilities for data management, and perhaps uniquely among statistical packages uses a binary data set format whose files are exchangeable among all platforms on which Stata is supported. Since converting data from ASCII into the native format is one of the slowest operations in any statistical package, it is preferable to use the native format for all data management operations. The platform-independence of the binary format implies that, for instance, a Windows user may exchange binary datasets with a Macintosh or Linux user without any need for data set translation. Stata Corporation also resells Stat/Transfer (a product of Circle Systems, <http://www.circlesys.com>), which supports interchange between several versions of Stata and the binary formats of many other statistical packages and matrix languages.

Binary data set compatibility, coupled with Stata's built-in support of `http`, makes it possible for any copy of Stata with internet access to read a binary data set over the web, merely referencing its URL. The version 8 command `webuse` provides easy access to any of the dozens of data sets used to illustrate Stata usage in the documentation, available from <http://stata-press.com/data>. Stata also provides excellent support for “reshaping” of data sets: e.g. conversion of panel data from “long” to “wide” format, or collapsing them on one of their defining indicators, computing summary statistics as specified. Version 8 also contains support for ODBC database access on Windows, Macintosh and Linux platforms.

Stata's statistical capabilities include standard regression and analysis of variance procedures, with particularly well developed strengths in panel data estimation techniques (the `xt` commands), survey data estimation (the `svy` commands), survival-time modelling (the `st` commands), and a wide variety of limited dependent variable models. Although Stata's strengths have traditionally been centered on cross-section and panel data estimation techniques, a growing set of sophisticated time-series methods are available. These include ARIMA, including ARMA-X models; a number of ARCH models; and, new in version 8, vector autoregressions (VARs) and structural VARs, impulse response functions (IRFs), and a suite of graphics capabilities for the presentation of IRFs. Both user-specified nonlinear least squares and maximum likelihood estimation capabilities are provided, although no support for nonlinear systems estimation (e.g., FIML) is present. Those seeking to use the maximum likelihood capabilities are urged to acquire the Gould, Pitblado and Sribney book, *Maximum Likelihood Estimation with Stata*, which was published in a second edition in July 2003 by Stata Press. Robust standard errors (heteroskedasticity-consistent, or “White” standard errors) are available for almost every estimation command. Newey-West or “HAC” standard errors are

only available for univariate regression models. Almost unique among statistical packages, Stata provides support for “clustering” in many estimation contexts: the notion that observations may be members of clusters, and the error structure must take the intra-cluster and between-cluster correlations into account. Marginal effects and (semi-)elasticities, and their standard errors, may be very easily generated in most estimation contexts with the `mf` command.

As mentioned above, Stata version 8 provides a completely new graphics subsystem with very powerful control over all aspects of publication-quality graphs. At present, Stata graphics commands support only two-dimensional plots and do not include contour plots. However, its developers have indicated that the new graphics language enables the development of three-dimensional plots, surface graphs, and the like, and users might expect those capabilities to be forthcoming from the vendor or from enterprising users.

No discussion of Stata’s capabilities would be complete without mention of the vibrant and productive Stata user community. Although Stata is a commercial product, its developers have aggressively encouraged the development of user-written procedures extending the Stata language by making the same development tools that are used in-house available to Stata users. A user-written Stata procedure—that is, a formal program, declared as such and stored in an “ado-file” of the same name—is a first-class citizen in the Stata context. When a user-written program is placed on the `adopath` (a set of directories akin to a Unix, Linux or DOS `path`) will appear as a built-in command. If a help file is provided (as it should be for any well-written program), it will appear as on-line help and may be viewed with hyperlinks in the graphical user interface of Stata’s Viewer window. Thus, users may develop useful, general-purpose extensions to Stata, and when those files are appropriately installed, they will be accessible and well-documented for even a novice user.

The installation of user-written procedures is another major innovation of Stata’s design. Since Stata functions as a web browser, it may retrieve a set of programs and help files—in its parlance, a Stata “package”—across the web and place those files on the `adopath` by means of the `net install` command, available from the application’s Help menu. An archive of user-written procedures for Stata and other statistical packages and languages, the Boston College Statistical Software Components (SSC) Archive (<http://ideas.repec.org/s/boc/bocode.html>) makes this process even easier. Package authors who submit their materials to the SSC Archive may use the Archive to disseminate their packages from a single source, accessible from within any copy of Stata with internet access via the `ssc` command.² For instance, if users find that the `ivreg2` command (Baum et al., 2003) would be a useful tool for their research, they may issue the command `ssc describe ivreg2` to learn more about the command, read the help file, etc., and the command `ssc install`

²Stata also supports proxy servers for those users with corporate firewalls.

`ivreg2` to download and install the package. If that package’s authors announce updates (bug fixes or enhancements), the same command may be used with the `replace` option to download and install the new files.

Stata Corp’s support of the user community includes publication of a refereed journal, *The Stata Journal* (<http://stata-journal.com>), which is in its third year, replacing the *Stata Technical Bulletin*. Although the journal is published by Stata Corporation, the editorial board operates independently. Likewise, the active *Statalist* listserv (<http://www.stata.com/support/statalist/>) is operated independently and maintained by Prof. Marcello Pagano of the Harvard School of Public Health. Many authors of Stata programs use *Statalist* to announce the availability of new or modified routines, and users’ questions are often answered the same day. StataCorp developers, including the president of the company, are frequent contributors to the discussion.

Stata’s web-accessibility extends to the maintenance of the software itself, which is comprised of two components: a relatively small (4 Mb) executable and a set of several hundred ancillary files that provide the majority of Stata commands’ definition, help files, dialog boxes, and graphics “schemes”. If the user purchases a perpetual license, he or she has the right to use that version of Stata without further maintenance fees, and all updates for that version are freely downloadable. The command `update query` will allow the user to determine whether new versions of the executable and/or ancillary files are available on Stata’s server. If so, the version may be updated and automatically installed with a single additional command. This permits the vendor to ensure that all users have access to the latest bug fixes and command enhancements. The vendor’s attention to any reported difficulties is exemplary; the identification of any flaw in the program in an exchange on *Statalist* often brings an immediate response from the developers that a fix is being tested and will be made accessible within a few days. Many of the updates also include significant enhancements to existing commands, or new commands, in response to user feedback.

4 Numerical reliability of Stata

4.1 Benchmark results

The URL <http://www.stata.com/support/cert/> provides links to pages indicating Stata’s performance in various benchmark suites. The NIST StRD pages report on tests for univariate summary statistics, linear regression, analysis of variance and nonlinear regression. Results are presented for Stata 8.1 running in a RedHat Linux environment and may differ slightly on other

platforms; the vendor states that the same numerical code is employed on all platforms. All tests but one in the NiST suite were completed with reasonable numbers of significant digits indicated. In the remaining test, Stata dropped collinear variables from the regression, an acceptable result. Detailed results for each test, as well as the code required to run the test and output log, are available from the web site.

The above URL also provides a link to DIEHARD benchmark results, developed by George Marsaglia, for uniform random-number generation. The results of the tests are indicated, as well as the complete methodology used to produce them. A technical note in the *Reference Manual* article on `functions` indicates that Stata’s pseudo-random-number generator is based on Marsaglia’s KISS generator and gives details of its algorithm.

4.2 Methodologies

The internal workings of a number of estimation commands that are often sources of numerical inaccuracy are open to view in Stata, and their descriptions in the *Reference Manuals* provide a reasonably detailed explanation of the methodologies employed. The source code for any routine for which the manual states “implemented as an ado-file” is accessible to the user. Alternatively, the enquiry `which command` will indicate either a built-in command (like `regress`), and thus not accessible, or it will provide the location on disk in which the command’s readable code resides.

For instance, the `arma` command is used to estimate Box–Jenkins models as well as related models, such as regression models with ARMA errors. Several pages of the *Time-Series Reference Manual* and two dozen references to the scientific literature provide considerable detail on the methods used to generate estimates with `arma`. On the other hand, the documentation does not describe the numerical methods used to generate ordinary least squares regression estimates, beyond presenting the appropriate linear algebraic representation.

The nonlinear least squares routine, command `nl`, employs a modified Gauss–Newton approach and does not provide alternative solution methods. However, since its code is freely available, users could extend this routine if they so chose. The maximum-likelihood routines, provided by the `ml` family of commands, provide a choice of several well-known solution methods (the maximum-likelihood-based `arch` and `arma` commands automatically switch among solution methods as needed). Detailed logs of the progress of estimation may be generated. A model to be estimated by maximum likelihood may be defined without derivatives (method `d0`), with first derivatives (method `d1`) or with second derivatives (method `d2`). There is no automatic generation of analytic derivatives. As

a unique feature, the general support for robust standard errors and clustering within Stata extends to user-specified models estimated by maximum likelihood.

5 Time series analysis with Stata

Although Stata is a relatively recent entrant to the time-series field, its developers have added significant capabilities in each of the last three releases. The `tsset` command allows the specification of a calendar variable which must be an integer counter. The `tsmktim` routine (Baum and Wiggins, 2001) allows such a variable to be created and its frequency defined in one step; e.g., `tsmktim yearqtr, start(1979q2)` will define the variable `yearqtr` in the current workspace, issue the `tsset` command and specify that the data are to be treated as quarterly. The `tin()` function may then be used to define a range of dates over which data transformations or statistical analysis is to take place: e.g., `regress profits sales trend if tin(1982q1,1989q4)` would run the regression over that sample. A forecast from the model for the 1990s may then be generated by `predict profhat, xb`. The `predict` command will, by default, compute both in-sample and out-of-sample predictions if the latter data for the regressors are available. A number of ancillary measures may be calculated by `predict`, depending on the estimation technique used. For OLS regression, these measures include fitted values, residuals, standardized residuals, studentized (jackknifed) residuals, standard errors of prediction, forecast and residual, and leverage estimates. Several additional residual diagnostic measures may be computed for the estimation sample.

The utility of `tsset` extends to the definition of panel data; for instance, `tsset firm year` may be used to define the data as a panel, and will automatically sort the data into that order (by firm, by year within firm). This implies that unbalanced panels are readily handled. In either the single time-series or panel context, the time-series operators are invaluable. These operators include `L.` for lag, `F.` for lead, and `D.` for difference. They may be combined, and may take a numeric argument or range: for example, `L(1/6).x` will define a list of the first six lags of `x` and may be written as such in a regression variable list. The construct `L3D2.y` would refer to the third lag of the second difference of the variable `y`. Most importantly, use of these operators, which is only allowed when the data have been `tsset`, prevents errors that might arise with explicit subscripting. For instance, is `x[_n-1]` last quarter's value of `x`? If there are missing observations, it could refer to a gap of two quarters. In that example, `L.x` will be defined as missing if the prior quarter's value is missing. Likewise, in a panel context, a lag, lead or difference will never extend into another unit's time series. However, these advantages do not come without their costs. Stata handles yearly, semiannual, quarterly, monthly, weekly and daily formats, but does not have support for

business–daily data. Many of the time–series commands refuse to operate when there are gaps in the time series, requiring the researcher to deal with holidays and other missing data values prior to estimation. The handling of weekly data can also be problematic. However, one may always define an integer variable from the observation number, and define the time–series calendar according to that variable, which circumvents many of these difficulties.

5.1 An overview of time series features

Stata’s *Time–Series Reference Manual* presents three categories of commands for use with univariate time series: estimators, diagnostic tools, and smoothers and filters. The estimators include `arima`, `arch`, `newey` and `prais`. The latter command estimates regressions with AR(1) errors via Prais–Winsten or Cochrane–Orcutt methods, while `newey` estimates regression models with Newey–West (HAC) standard errors. The `arima` command, as mentioned above, is capable of estimating a variety of models beyond the standard Box–Jenkins approach, such as a regression model with ARMA disturbances. Several alternative optimization methods (such as BFGS, DFP and BHHH) are provided, and may be specified by the user. The predictions from `arima` may be specified as one–step–ahead or dynamic, and they may refer to the level of the series even if the model was fit on the differences. Stata’s implementation of `arch` includes ARCH, GARCH, ARCH–in–mean, Nelson’s EGARCH, threshold ARCH, and several forms of nonlinear ARCH. Several forms of predictions are available, including the predicted conditional variance, both in– and out–of–sample. Stata’s built–in `constraint` facility may be used to constrain ARCH coefficients; an example in the manual reproduces Engle’s original estimates in which an arithmetic lag is imposed on the ARCH coefficients.

Diagnostic tools available for univariate time series include a number of unit root tests, several frequency domain measures and tests for white noise and ARCH effects. The Durbin–Watson test is also included in this category; the D–W statistic is not automatically provided by `regress`. Smoothers available in Stata include exponential, double exponential and Holt–Winters smoothing, both seasonal and nonseasonal. Moving average and nonlinear filters are also provided.

Multivariate time–series techniques described in the *Time–Series Reference Manual* all relate to vector autoregressions and the broad suite of capabilities for those models. This includes standard VAR estimation, structural VAR estimation, and the generation of diagnostic tests (such as Granger causality and lag–order selection statistics), dynamic forecasts, forecast–error variance decompositions and impulse response functions in point and interval form.

Other techniques relevant to time series modelling are available in Stata. For instance, the cross-section time-series capabilities include `xtabond`, the original form of the Arellano–Bond dynamic panel data estimator, `xtregar`, fixed- and random-effects linear models with an AR(1) disturbance and `sureg`, seemingly unrelated regression models. The `reg3` command provides three-stage least squares estimation of systems of simultaneous equations. Although Stata has a quite powerful bootstrapping facility, it does not offer support for bootstrapping after time-series estimation nor the specification of a block bootstrap.

5.2 An illustration of time series analysis

The following Stata 8.1 program illustrates some of the estimation, forecasting and graphics features of the package. A few explanatory points: (1) commands to create a new variable (`generate`) or modify an existing variable automatically operate on the entire dataset unless otherwise directed; no explicit loops over observations are required, and are discouraged as highly inefficient, as in most matrix languages. (2) the `local` construct (and `global`, unused here) is a single object which may hold either numeric or string values. The expression ‘`localname`’ dereferences the object. Stata also contains the `scalar` and `matrix` constructs, not used here. (3) The `graph export` as PDF is only available for Macintosh. As noted above, other export formats are available, including Windows-specific formats.

In this program, modified from the `arima` entry of the *Time-Series Reference Manual*, we analyse quarterly US wholesale price index data, computing log differences (see Figure 1) and examining the autocorrelogram and partial autocorrelogram (see Figure 2). Enders (1995) illustrates ARIMA modelling with these data, using these diagnostics to choose a model including an AR(1) term, a MA(1) term, and a MA(4) term to deal with a remaining quarterly effect. The resulting model is:

$$\Delta \ln(wpi_t) = \beta_0 + \rho \{ \Delta \ln(wpi_{t-1}) - \beta_0 \} + \theta_1 \epsilon_{t-1} + \theta_4 \epsilon_{t-4} + \epsilon_t \quad (1)$$

By default, outer-product-of-gradient estimates of the parameter covariance matrix are computed, as displayed below. We could have specified that the covariance matrix should be derived from the Hessian of the likelihood function. Following estimation over the sample ending in 1985Q4, the out-of-sample observations for 1986–1990 are used to generate dynamic forecasts and comparisons between actual and forecast inflation over that period (see Figure 3).

The rapid evolution characteristic of Stata’s development is evidenced by the program below, which

incorporates a graphics command, `tsline`, which was added to official Stata on 12 September 2003. This new command makes production of time-series graphs with appropriate date labels much simpler. The command is fully documented in on-line help in an updated version of Stata 8.1.

```

log using tsdemo.log, replace
* Data from Enders, Applied Econometric Time Series, 1995.

set scheme s2mono

webuse wpi1, clear
label var wpi "US Wholesale Price Index (WPI)"
gen double dlnwpi = D.ln_wpi
label var dlnwpi "US WPI -- log difference"

* graph the series and its log difference
tsline wpi , tlabels(#7) ylabel(, angle(horizontal)) name(fig1a)
tsline dlnwpi, tlabels(#7) ylabel(, angle(horizontal)) name(fig1b)
graph combine fig1a fig1b, col(1) ysize(6.5) xsize(8.5) title(Figure 1)
graph export fig1.pdf, replace

* plot the autocorrelogram and partial autocorrelogram
ac dlnwpi, ylabel(-.4(.2).6, angle(horizontal)) name(fig2a)
pac dlnwpi, ylabel(-.4(.2).6, angle(horizontal)) name(fig2b)
graph combine fig2a fig2b, col(1) iscale(0.6) ysize(6.5) xsize(8.5) ///
    title(Figure 2)
graph export fig2.pdf, replace

* Fit an ARIMA(1,1,1) model to the series, with an additional 4th order MA term
arima dlnwpi if tin(,1985q4), ar(1) ma(1 4)

* Generate dynamic forecasts for 1986-1990
local exante tin(1986q1,1990q4) // used many times
predict double dlnwpihat if 'exante', xb dynamic(.)
gen fcerr = dlnwpi - dlnwpihat
label var fcerr "Forecast error from inflation predictions"

* plot the actual, predicted, and forecast error series
tsline dlnwpi dlnwpihat if 'exante', ///
    ylabel(, angle(horizontal) format(%4.2f)) yline(0) ytitle("") ///
    title("Out-of-sample dynamic forecasts for US WPI inflation, 1986-1990") ///
    name(fig3a)
tsline fcerr if 'exante', ///
    ylabel(, angle(horizontal) format(%4.2f)) yline(0) ytitle("") ///
    title(Forecast errors) recast(area) name(fig3b)
graph combine fig3a fig3b, col(1) iscale(0.6) ysize(6.5) xsize(8.5) ///
    title(Figure 3)
graph export fig3.pdf, replace
log close

```

Estimation results are given as follows. Stata's default display format uses `SMCL`, Stata's own

markup language, to provide more elegant output than the typewriter-format text log excerpted here. The SMCL logs also include colored elements, and may be readily transformed into HTML for web presentation.

ARIMA regression

```
Sample: 1960q2 to 1985q4                Number of obs   =      103
                                         Wald chi2(3)    =      243.00
Log likelihood = 325.0663                Prob > chi2     =      0.0000
```

		OPG				[95% Conf. Interval]	
		Coef.	Std. Err.	z	P> z		
dlnwpi							
dlnwpi	_cons	.0107584	.0063455	1.70	0.090	-.0016785	.0231952
ARMA							
ar	L1	.7738686	.0971293	7.97	0.000	.5834987	.9642386
ma	L1	-.375707	.1276934	-2.94	0.003	-.6259814	-.1254327
	L4	.353606	.1292881	2.74	0.006	.1002059	.607006
	/sigma	.0102384	.000472	21.69	0.000	.0093133	.0111636

6 Summary

Within the last several releases, Stata has evolved from a package with strengths in cross-sectional and panel data handling to a more general purpose tool with sophisticated capabilities for ARCH, ARIMA and VAR modelling. Although its support for a variety of common time series methods is still under development, I feel quite comfortable recommending Stata to first-year economics and finance doctoral students and suggesting that it will meet a very sizable percentage of their research needs. The addition of publication-quality graphics makes it possible to depend on one package rather than having to export data to other graphics production packages, and the introduction of a menu-driven interface makes Stata accessible to even the undergraduate statistics student with no prior computing experience. Although no single package can serve all needs, Stata's developers have done an admirable job of listening to their users and vigorously responding to commonly-evoked needs. Following that course, the package will only become more valuable and useful to researchers in a diverse set of disciplines.

References

- [1] Baum, Christopher F., Mark E. Schaffer, and Steven Stillman, 2003. Instrumental variables and GMM: Estimation and testing. *Stata Journal*, 3, 1–31.
- [2] Baum, Christopher F. and Vince Wiggins, 2001. Utility for time series data. *Stata Technical Bulletin Reprints*, Vol. 10, 29–30. College Station, TX: Stata Corporation.
- [3] Berk, Kenneth N., 1987. Effective Microcomputer Statistical Software. *The American Statistician*, 41, 222–228.
- [4] Enders, Walter, 1995. *Applied Econometric Time Series*. New York: John Wiley & Sons.
- [5] Kolenikov, Stanislav, 2001. Review of Stata 7. *Journal of Applied Econometrics*, 16, 637–646.

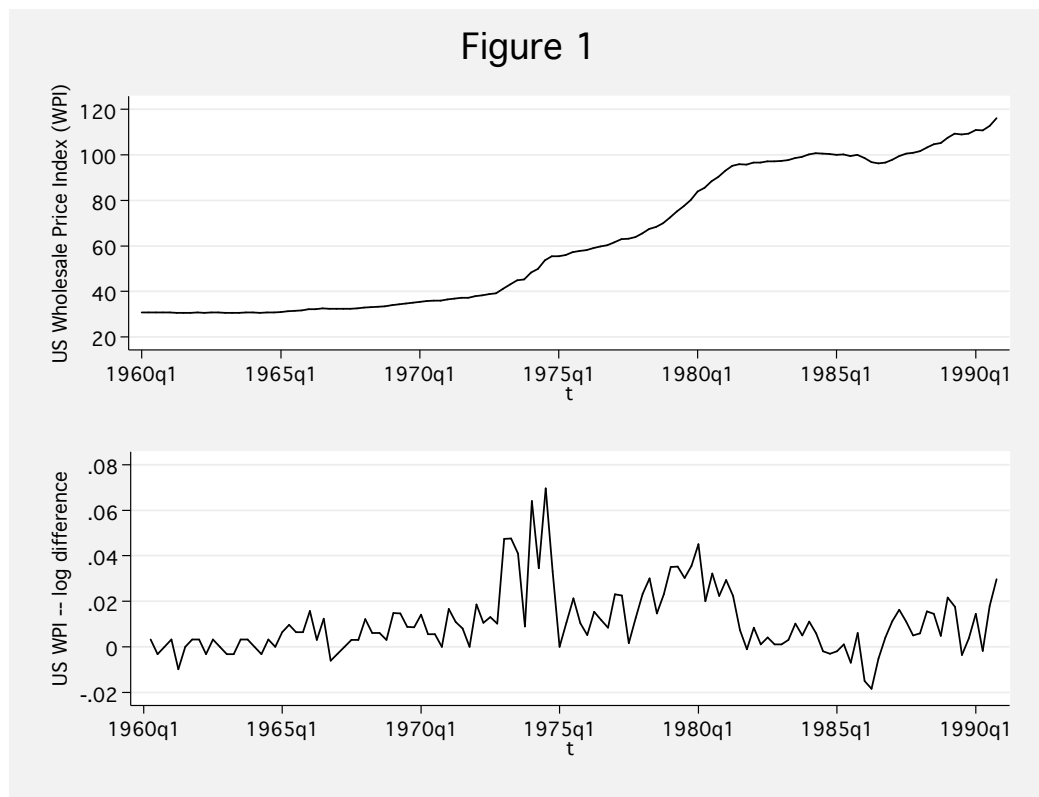


Figure 2

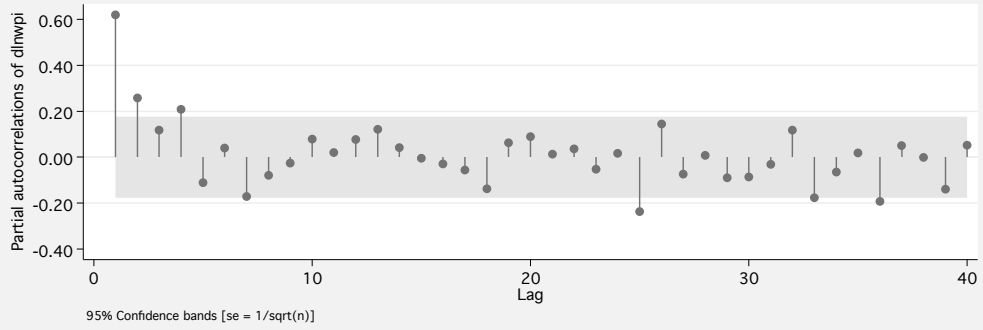
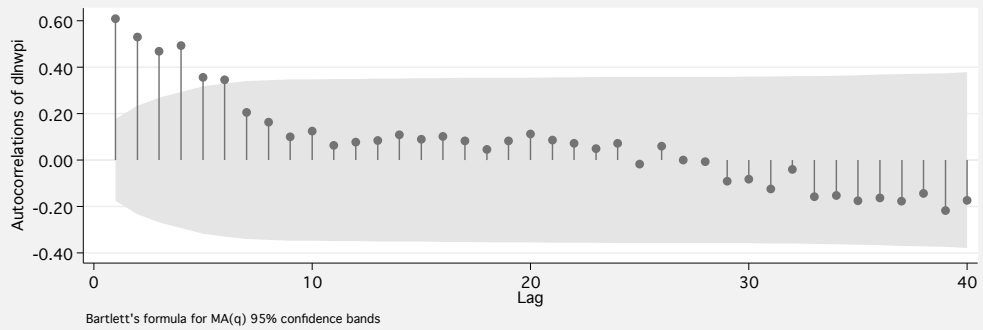
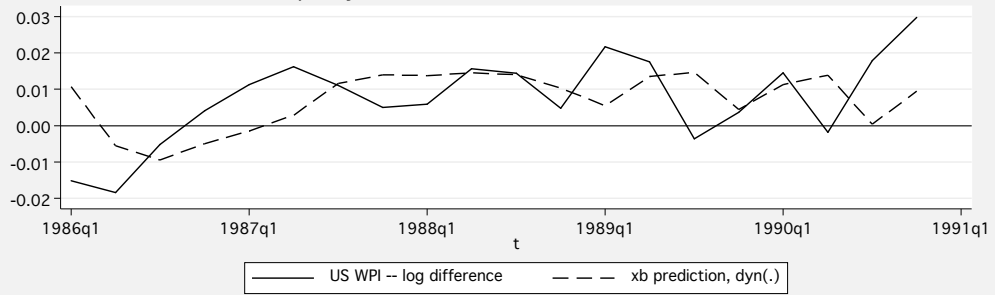


Figure 3

Out-of-sample dynamic forecasts for US WPI inflation, 1986-1990



Forecast errors

