The *Stata Journal* publishes reviewed papers together with shorter notes or comments, regular columns, book reviews, and other material of interest to Stata users. Examples of the types of papers include 1) expository papers that link the use of Stata commands or programs to associated principles, such as those that will serve as tutorials for users first encountering a new field of statistics or a major new technique; 2) papers that go "beyond the Stata manual" in explaining key features or uses of Stata that are of interest to intermediate or advanced users of Stata; 3) papers that discuss new commands or Stata programs of interest either to a wide spectrum of users (e.g., in data management or graphics) or to some large segment of Stata users (e.g., in survey statistics, survival analysis, panel analysis, or limited dependent variable modeling); 4) papers analyzing the statistical properties of new or existing estimators and tests in Stata; 5) papers that could be of interest or usefulness to researchers, especially in fields that are of practical importance but are not often included in texts or other journals, such as the use of Stata in managing datasets, especially large datasets, with advice from hard-won experience; and 6) papers of interest to those who teach, including Stata with topics such as extended examples of techniques and interpretation of results, simulations of statistical concepts, and overviews of subject areas.

The *Stata Journal* is indexed and abstracted by *CompuMath Citation Index*, *Current Contents/Social and Behavioral Sciences*, *RePEc: Research Papers in Economics*, *Science Citation Index Expanded* (also known as *SciSearch*, *Scopus*, and *Social Sciences Citation Index*.

For more information on the *Stata Journal*, including information for authors, see the webpage

http://www.stata-journal.com

Subscriptions are available from StataCorp, 4905 Lakeway Drive, College Station, Texas 77845, telephone 979-696-4600 or 800-STATA-PC, fax 979-696-4601, or online at

http://www.stata.com/bookstore/sj.html

Subscription rates listed below include both a printed and an electronic copy unless otherwise mentioned.

U.S. and Canada		Elsewhere	
1-year subscription	$ 79	1-year subscription	$115
2-year subscription	$155	2-year subscription	$225
3-year subscription	$225	3-year subscription	$329
3-year subscription (electronic only)	$210	3-year subscription (electronic only)	$210
1-year student subscription	$ 48	1-year student subscription	$ 79
1-year university library subscription	$ 99	1-year university library subscription	$135
2-year university library subscription	$195	2-year university library subscription	$265
3-year university library subscription	$289	3-year university library subscription	$395
1-year institutional subscription	$225	1-year institutional subscription	$259
2-year institutional subscription	$445	2-year institutional subscription	$510
3-year institutional subscription	$650	3-year institutional subscription	$750

Back issues of the *Stata Journal* may be ordered online at

http://www.stata.com/bookstore/sjj.html

Individual articles three or more years old may be accessed online without charge. More recent articles may be ordered online.

http://www.stata-journal.com/archives.html

The *Stata Journal* is published quarterly by the Stata Press, College Station, Texas, USA.

Address changes should be sent to the *Stata Journal*, StataCorp, 4905 Lakeway Drive, College Station, TX 77845, USA, or emailed to sj@stata.com.

The *Stata Journal* (ISSN 1536-867X) is a publication of Stata Press. Stata, STATA, Stata Press, Mata, MATA, and NetCourse are registered trademarks of StataCorp LP.

The Stata Journal (2012)
12, Number 3, pp. 515–542

Long-run covariance and its applications in cointegration regression

Qunyong Wang
Institute of Statistics and Econometrics
Nankai University
Tianjin, China
brynewqy@nankai.edu.cn

Na Wu
Economics School
Tianjin University of Finance and Economics
Tianjin, China

Abstract. Long-run covariance plays a major role in much of time-series inference, such as heteroskedasticity- and autocorrelation-consistent standard errors, generalized method of moments estimation, and cointegration regression. We propose a Stata command, `lrcov`, to compute long-run covariance with a prewhitening strategy and various kernel functions. We illustrate how long-run covariance matrix estimation can be used to obtain heteroskedasticity- and autocorrelation-consistent standard errors via the new `hacreg` command; we also illustrate cointegration regression with the new `cointreg` command. `hacreg` has several improvements compared with the official `newey` command, such as more kernel functions, automatic determination of the lag order, and prewhitening of the data. `cointreg` enables the estimation of cointegration regression using fully modified ordinary least squares, dynamic ordinary least squares, and canonical cointegration regression methods. We use several classical examples to demonstrate the use of these commands.

Keywords: st0272, lrcov, hacreg, cointreg, long-run covariance, fully modified ordinary least squares, dynamic ordinary least squares, canonical cointegration regression

1 Introduction

Long-run covariance (LRCOV) plays a major role in much of time-series inference, such as heteroskedasticity- and autocorrelation-consistent (HAC) standard errors, efficient generalized method of moments (GMM) estimation, cointegration regression, etc. Asymptotic theory for estimators has developed rapidly within the literature, primarily focused on the LRCOV. Practical applications of robust inference that take into account potential heteroskedasticity and autocorrelation of unknown forms in the data involve LRCOV matrix estimation, such as White heteroskedasticity robust standard errors and Newey–West HAC standard errors.

Unit-root tests and cointegration tests are now routinely used in empirical research. LRCOV has been widely applied to nonstationary time-series analysis, such as the

Phillips–Perron unit-root test (Phillips and Perron 1988), cointegration tests (Marmol and Velasco 2004) and panel cointegration tests (Pedroni 2004), a model's stability based on fully modified ordinary least squares (FMOLS) (Hansen 1992), canonical correlation regression (CCR) with both $I(1)$ and $I(2)$ variables (Choi, Park, and Yu 1997), fully modified value at risk (VAR), and fully modified GMM estimation (Quintos 1998).

Three approaches are popular for computing LRCOV: the nonparametric kernel method, the parametric method, and the prewhitened kernel method. The kernel method has a long history for which the kernel and bandwidth are two important determinants of the finite-sample properties of LRCOV. Many kernels have been proposed (Priestley 1981), among which the Bartlett, Parzen, and quadratic spectral may be the most popular choices. Some new kernels have been proposed recently, such as the class of steep-origin kernels of Phillips, Sun, and Jin (2007).

To date, there are two widely used formulas for selecting the bandwidth, proposed in Andrews (1991) and Newey and West (1994). Hirukawa (2010) proposed an alternative approach, namely, the two-stage plug-in bandwidth selection approach. For the parametric approach, econometric models are used to prewhiten the data; these models include the VAR prewhitening of den Haan and Levin (1997) and the autoregressive moving-average (ARMA) prewhitening of Lee and Phillips (1994). The prewhitened kernel method (Andrews and Monahan 1992) combines the parametric method and the nonparametric kernel method. All three approaches will be discussed in section 2.

Some software is able to compute LRCOV and fit some relevant models with its applications (software such as EViews, Rats, and the Coint package of Gauss). The time-series utilities of Stata have increased rapidly. Some existing Stata commands estimate LRCOV implicitly (`newey`, `ivreg2`, etc.), but there is still no explicit command available.

In this article, we offer the `lrcov` command for computing the symmetric and one-sided LRCOV in Stata. The `lrcov` program supports Andrews (1991) and Newey and West (1994) automatic bandwidth selection methods for kernel estimators, as well as information criteria–based lag-length selection methods for value-at-risk HAC (VARHAC) and prewhitening estimation. We also provide two additional commands, `hacreg` and `cointreg`, as applications of `lrcov`. `hacreg` estimates HAC standard error and extends Stata's official `newey` command in some ways. `cointreg` estimates FMOLS, dynamic ordinary least squares (DOLS), and CCR.

The remainder of the article is organized as follows. The background of LRCOV is introduced in section 2. In section 3, we introduce the syntax of `lrcov` and its applications in the calculation of HAC standard error and GMM estimation. `hacreg` and `cointreg` are illustrated in sections 4 and 5. In section 6, we draw conclusions and outline future research.

2 Background of LRCOV

For second-order stationary processes, the long-run variance is defined as the sum of all autocovariances or, equivalently, in terms of the spectrum at frequency zero. Consider a sequence of mean-zero random p-vectors, $\mathbf{v}_t(\theta)$, that have K parameters. The LRCOV matrix $\mathbf{\Omega}$ of $\mathbf{v}_t(\theta)$ is

$$\begin{aligned} \mathbf{\Omega} &= \sum_{j=-\infty}^{\infty} \mathbf{\Gamma}_j \\ \mathbf{\Gamma}_j &= E(\mathbf{v}_t \mathbf{v}'_{t-j}), \quad j \geq 0 \\ \mathbf{\Gamma}_j &= \mathbf{\Gamma}'_{-j}, \qquad\quad\; j < 0 \end{aligned}$$

where $\mathbf{\Gamma}_j$ is the autocovariance matrix of $\mathbf{v}_t$ at lag j. Define the one-sided LRCOV $\mathbf{\Lambda}_0$ and strict one-sided LRCOV $\mathbf{\Lambda}_1$ as

$$\begin{aligned} \mathbf{\Lambda}_0 &= \sum_{j=0}^{\infty} \mathbf{\Gamma}_j = \mathbf{\Lambda}_1 + \mathbf{\Gamma}_0 \\ \mathbf{\Lambda}_1 &= \sum_{j=1}^{\infty} \mathbf{\Gamma}_j \end{aligned}$$

where $\mathbf{\Gamma}_0$ is the contemporaneous covariance. So the symmetric two-sided LRCOV can also be written as

$$\mathbf{\Omega} = \mathbf{\Lambda}_1 + \mathbf{\Lambda}_1{}' + \mathbf{\Gamma}_0 = \mathbf{\Lambda}_0 + \mathbf{\Lambda}_0{}' - \mathbf{\Gamma}_0$$

Given the definition of $\mathbf{\Omega} = \sum_{j=-\infty}^{\infty} \mathbf{\Gamma}_j$, it is natural to estimate $\mathbf{\Omega}$ using the sample autocovariances, $\widehat{\mathbf{\Gamma}}_j = T^{-1} \sum_{t=j+1}^{T} \mathbf{v}_t \mathbf{v}'_{t-j}$, as estimates of their population analogs. This leads to the one-sided estimator

$$\widehat{\mathbf{\Omega}} = \widehat{\mathbf{\Gamma}}_0 + \sum\nolimits_{j=1}^{K} \widehat{\mathbf{\Gamma}}_j$$

White and Domowitz (1984) first proposed this type of estimator and showed its consistency. While $\widehat{\mathbf{\Omega}}$ converges in probability to a positive-definite matrix, it may be indefinite in finite samples. As argued by Hall (2005), the source of the trouble lies in the weights given to the sample autocovariances. The solution is to construct an estimator in which the contributions of the sample autocovariance matrices are weighted to downgrade their role sufficiently in finite samples to ensure positive semidefiniteness; however, the weights also need to tend to one as $T \to \infty$ to ensure consistency. This is the intuition behind the nonparametric kernel approach of HAC matrices (Andrews 1991; Newey and West 1987). The nonparametric kernel approach estimates the LRCOV by taking a weighted sum of the sample autocovariances of the observed data

$$\widehat{\mathbf{\Omega}} = \widehat{\mathbf{\Gamma}}_0 + \sum\nolimits_{j=1}^{K} k(j) \widehat{\mathbf{\Gamma}}_j$$

where $k(j)$ is known as the kernel (or weight). The kernel must be chosen to ensure the twin properties of consistency and positive semidefiniteness.

Below we will discuss the nonparametric kernel method and two other methods to estimate $\mathbf{\Omega}$, that is, the parametric VARHAC approach (den Haan and Levin 1997) and the prewhitened kernel approach (Andrews and Monahan 1992).

2.1 Nonparametric kernel method

The class of kernel HAC covariance matrix estimators in Andrews (1991) may be written as

$$\begin{aligned} \widehat{\mathbf{\Omega}} &= \frac{T}{T-K} \sum_{j=-\infty}^{\infty} k(j/b_T)\widehat{\mathbf{\Gamma}}_j \\ \widehat{\mathbf{\Gamma}}_j &= \frac{1}{T}\sum_{t=j+1}^{T} \mathbf{v}_t\mathbf{v}'_{t-j}, \qquad j \geq 0 \\ \widehat{\mathbf{\Gamma}}_j &= \widehat{\mathbf{\Gamma}}'_{-j}, \qquad j < 0 \end{aligned} \tag{1}$$

k is a symmetric kernel (or lag window) function that is continuous at the origin and satisfies $k(x) \leq 1$ and $k(0) = 1$. The allowed kernels in `lrcov` are listed in table 1. Candidate kernel functions can be found in standard texts, for example, Brillinger (1980) and Priestley (1981). b_T is the bandwidth parameter, which depends on the number of observations. b_T controls the number of autocovariances included in the LRCOV estimator for some kernels, such as the Bartlett and Parzen kernels. $T/(T-K)$ is an optional correction of degrees of freedom associated with the parameters in the model. So the nonparametric kernel estimator can be viewed as a weighted average of sample autocovariances. The weights are just values of the lag window function evaluated at different lag lengths.

Andrews (1991) shows that the quadratic spectral weights are optimal in the sense that they minimize an asymptotic mean-squared-error criterion for the estimation of LRCOV. His results imply that this choice only marginally dominates the Parzen weights, but Parzen and quadratic spectral weights should be much better than the Bartlett weights. However, neither dominates the Bartlett weight to the extent predicted by the theory. Newey and West (1994) conclude that the choice between the kernels is not particularly important and the bandwidth is a much more important determinant of the finite-sample properties of LRCOV. For consistency, b_T must tend to infinity with T. Andrews (1991) shows that the asymptotic mean squared error is minimized by setting b_T equal to $O(T^{1/3})$ for the Bartlett weights and equal to $O(T^{1/5})$ for both the Parzen and quadratic spectral weights. However, this type of condition provides little practical guidance.

Table 1. Kernel function properties

Kernel	Function	c_k	q	*rate*
Bartlett	$k(x) = \begin{cases} 1 - \lvert x \rvert & \text{if} \quad x \leq 1.0 \\ 0 & \text{otherwise} \end{cases}$	1.1447	1	2/9
Bohman	$k(x) = \begin{cases} (1 - \lvert x \rvert)\cos(\pi x) + \frac{\sin(\pi \lvert x \rvert)}{\pi} & \text{if} \quad x \leq 1.0 \\ 0 & \text{otherwise} \end{cases}$	2.4202	2	4/25
Daniell	$k(x) = \sin(\pi x)/(\pi x)$	0.4462	2	–
Parzen	$k(x) = \begin{cases} 1 - 6x^2(1 - \lvert x \rvert) & \text{if} \quad 0 \leq \lvert x \rvert \leq 0.5 \\ 2(1 - \lvert x \rvert)^3 & \text{if} \quad 0.5 < \lvert x \rvert \leq 1.0 \\ 0 & \text{otherwise} \end{cases}$	2.6614	2	4/25
Parzen–Riesz	$k(x) = \begin{cases} 1 - x^2 & \text{if} \quad x \leq 1.0 \\ 0 & \text{otherwise} \end{cases}$	1.1340	2	4/25
Parzen–Geometric	$k(x) = \begin{cases} 1/(1 + \lvert x \rvert) & \text{if} \quad x \leq 1.0 \\ 0 & \text{otherwise} \end{cases}$	1.0000	1	2/9
Parzen–Cauchy	$k(x) = \begin{cases} 1/(1 + x^2) & \text{if} \quad x \leq 1.0 \\ 0 & \text{otherwise} \end{cases}$	1.0924	2	4/25
Quadratic spectral	$k(x) = \frac{25}{12\pi^2 x^2}\left\{\frac{\sin(1.2\pi x)}{1.2\pi x} - \cos(1.2\pi x)\right\}$	1.3221	2	2/25
Tukey–Hamming	$k(x) = \begin{cases} 0.54 + 0.46\cos(\pi x) & \text{if} \quad x \leq 1.0 \\ 0 & \text{otherwise} \end{cases}$	1.6694	2	4/25
Tukey–Hanning	$k(x) = \begin{cases} 0.50 + 0.50\cos(\pi x) & \text{if} \quad x \leq 1.0 \\ 0 & \text{otherwise} \end{cases}$	1.7462	2	4/25
Tukey–Parzen	$k(x) = \begin{cases} 0.436 + 0.564\cos(\pi x) & \text{if} \quad x \leq 1.0 \\ 0 & \text{otherwise} \end{cases}$	1.8576	2	4/25
Truncated uniform	$k(x) = \begin{cases} 1 & \text{if} \quad \lvert x \rvert \leq 1.0 \\ 0 & \text{otherwise} \end{cases}$	0.6611	1/5	–

Note: c_k and q are used to compute optimal bandwidth; *rate* is the optimal rate of increase for the lag selection in Newey and West (1987).

Andrews (1991) and Newey and West (1994) offer two automatic bandwidth selection techniques based on observations to estimating b_T. Both methods estimate b_T according to the following rule

$$\widehat{b}_T = c_k\{\widehat{\alpha}(q)T\}^{1/(2q+1)}$$

where c_k and q depend on the type of kernel function and are also listed in table 1. The Andrews (1991) method estimates b_T parametrically by fitting a simple parametric time-series model to the original data, and then deriving the autocovariances and corresponding $\alpha(q)$. For the univariate autoregressive AR(1) models corresponding to the p variables,

$$\widehat{\alpha}(q) = \frac{\sum_{s=1}^{p} w_s \left(\widehat{f}_s^q\right)^2}{\sum_{s=1}^{p} w_s \left(\widehat{f}_s^0\right)^2}$$
$$\widehat{f}_s^q = \frac{1}{2\pi} \sum_{j=-\infty}^{\infty} |j|^q \widetilde{\Gamma}_{s,j} \tag{2}$$

where $\widetilde{\Gamma}_{s,j}$ are the estimated autocovariances at lag j implied by the univariate AR(1) specification for the sth variable. w_s is the weight for the sth variable. Andrews (1991) suggests using either $w_s = 1$ for all or for all but the instrument corresponding to the intercept in regression settings. However, to date, no further guidance is available about how this choice should be made or its impact on the finite-sample properties of LRCOV.

Newey and West (1994) use a nonparametric approach. First, define the scalar autocovariance estimators

$$\widehat{\sigma}_j = \frac{1}{T} \sum_{t=j+1}^{T} \mathbf{w}' \mathbf{v}_t \mathbf{v}_{t-j}' \mathbf{w} = \mathbf{w}' \widehat{\mathbf{\Gamma}}_j \mathbf{w} \tag{3}$$

where $\mathbf{w} = (w_1, w_2, \dots, w_p)'$. Then compute nonparametric truncated kernel estimators of the Parzen measures of smoothness:

$$\widehat{f}_s^q = \frac{1}{2\pi} \sum_{j=-n}^{n} |j|^q \widehat{\sigma}_j \tag{4}$$

The Newey and West (1994) estimator for $\alpha(q)$ is

$$\widehat{\alpha}(q) = \left(\widehat{f}^q / \widehat{f}^0\right)^2$$

Kiefer and Vogelsang (2002a,b) proposed the use of inconsistent HAC estimates based on conventional kernels but with the bandwidth set equal to the sample size. They show that such estimates lead to asymptotically valid tests that can have better finite-sample-size properties than tests based on consistent HAC estimates. Their power analysis and simulations reveal that the Bartlett kernel produces the highest power function in regression testing with $b_T = T$, although power is noticeably less than what can be attained using conventional procedures involving consistent HAC estimators.

2.2 Parametric VARHAC method

One of the mechanisms that can improve covariance matrix estimates is derived from the idea of prewhitening. Prewhitening has a long history in time-series analysis. Recently, it has also been applied to the estimation of LRCOV. The stated idea of prewhitening is as follows. One first prefilters the original data to obtain a less dependent series that has a flatter spectrum. The spectral density function of the filtered data can then be estimated with less bias and recolored to produce an estimator for the spectrum of the original series with reduced bias. If the true data-generating process belongs to the parametric class that is used for prewhitening, then the parametric prewhitened estimator has improved the convergence rate.

den Haan and Levin (1997) proposed the VARHAC parametric method to estimate LRCOV. They first whiten the data using the VAR(q) model and compute the contemporaneous covariance for the whitened data. They then recolor it to the original data.

Let the filtered data be

$$\mathbf{v}_t^* = \mathbf{v}_t - \sum\nolimits_{j=1}^{q} \widehat{\mathbf{A}}_j \mathbf{v}_{t-j}$$

with contemporaneous covariance

$$\widehat{\boldsymbol{\Gamma}}_0^* = \frac{1}{T-q} \sum\nolimits_{t=q+1}^{T} \mathbf{v}_t^* \mathbf{v}_t^{*\prime}$$

The two-sided LRCOV is

$$\widehat{\boldsymbol{\Omega}} = \frac{T-q}{T-q-K} \widehat{\mathbf{D}} \widehat{\boldsymbol{\Gamma}}_0^* \widehat{\mathbf{D}}$$

$$\widehat{\mathbf{D}} = \left(\mathbf{I}_p - \sum\nolimits_{j=1}^{q} \widehat{\mathbf{A}}_j \right)^{-1}$$

where $\widehat{\boldsymbol{\Gamma}}_0^*$ is the contemporaneous covariance of $\mathbf{v}_t^*$. den Haan and Levin (1996) recommend choosing the lag length via a model-selection criterion. Their theoretical analysis suggests that the Bayesian information criterion is a better choice than Akaike's information criterion (AIC); however, their simulation evidence suggests that the two criteria perform comparably in this context. den Haan and Levin (1996) show that $\widehat{\boldsymbol{\Omega}}$ is consistent provided that $q \to \infty$ as $T \to \infty$ and $q = O(T^{1/3})$.

The VARHAC estimators for the one-sided LRCOV, $\boldsymbol{\Lambda}_0$ and $\boldsymbol{\Lambda}_1$, do not have simple expressions in terms of $\widehat{\mathbf{A}}$ and $\widehat{\boldsymbol{\Gamma}}_0^*$. For the VAR(1) model, the one-sided LRCOV may be written as (QMS 2010)

$$\widehat{\boldsymbol{\Lambda}}_0 = \frac{T-q}{T-q-K} \left(\mathbf{I}_p - \widehat{\mathbf{A}}_1 \right)^{-1} \widehat{\boldsymbol{\Gamma}}_0$$

$$\widehat{\boldsymbol{\Lambda}}_1 = \frac{T-q}{T-q-K} \widehat{\mathbf{A}}_1 \left(\mathbf{I}_p - \widehat{\mathbf{A}}_1 \right)^{-1} \widehat{\boldsymbol{\Gamma}}_0$$

where $\widehat{\boldsymbol{\Gamma}}_0$ can be computed using (1).

2.3 Prewhitened kernel method

The prewhitened kernel approach is a hybrid method that combines the parametric method and the nonparametric kernel method. The prewhitened method uses a parametric model to obtain residuals that prefilter the data and a nonparametric kernel estimator to obtain an LRCOV estimator of the whitened data. Andrews and Monahan (1992) proposed a VAR prewhitening procedure for the estimation of the covariance matrix. Lee and Phillips (1994) used the Hannan–Rissanen recursive estimation procedure and order-selection methods and proposed an ARMA prewhitened long-run variance estimator. All of these prewhitened estimators use parametric (AR or ARMA) prewhitening procedures in the time domain. The resulting prewhitened LRCOV estimate is then recolored to undo the effects of the transformation.

To estimate LRCOV using the method of Andrews and Monahan (1992), we first filter the data using VAR(q). We then construct the innovation's kernel LRCOV using (1). The formula of one-sided LRCOV for VAR(1) was given in Hansen (1992):

$$\widehat{\boldsymbol{\Lambda}}_0 = \widehat{\mathbf{D}}\widehat{\boldsymbol{\Lambda}}_0^*\widehat{\mathbf{D}}' + \widehat{\mathbf{D}}\widehat{\mathbf{A}}_1\widehat{\boldsymbol{\Gamma}}_0$$

The one-sided LRCOV for the VAR(q) model is derived by Park and Ogaki (1991):

$$\widehat{\boldsymbol{\Lambda}}_1 = \widehat{\mathbf{D}}\widehat{\boldsymbol{\Lambda}}_1^*\widehat{\mathbf{D}}' + \widehat{\mathbf{D}}\sum\nolimits_{j=0}^{q-1}\sum\nolimits_{i=j+1}^{q}\widehat{\mathbf{A}}_i\widehat{\boldsymbol{\Gamma}}_j$$

The VARHAC method is advantageous because the LRCOV can be estimated straightforwardly from the model. den Haan and Levin (2000) found that once data-dependent VAR prewhitening has been used in linear regression, the effect of the prewhitened kernel method is negligible or even counterproductive. The potential disadvantage is that if the model is incorrect, then the LRCOV estimator is inconsistent. The closer the prewhitened model is to the true model, the less bias in the estimator. Indeed, Andrews and Monahan (1992) exhibit examples where the mean squared error can actually be worse than that of the standard estimator. Sul, Phillips, and Choi (2005) showed that the small-sample bias in the estimation of autoregressive coefficients is transmitted to the recoloring filter, leading to HAC variance estimates that can be badly biased. They recommended using recursive demeaning procedures to mitigate the effects of small-sample autoregressive bias.

For the kernel method, the LRCOV estimator is consistent under much weaker conditions than the VARHAC method. Unfortunately, these more general estimators can exhibit poor finite-sample performance, which prompted the construction of prewhitened kernel estimator.

As argued by Andrews and Monahan (1992) and Lee and Phillips (1994), the prewhitened kernel estimator of the LRCOV reduces bias and improves the rate of convergence of existing estimators. Simulation evidence in Newey and West (1994) suggests that the use of prewhitening and recoloring improves the finite-sample performance of the parameters' asymptotic confidence intervals. Xiao and Oliver (2002) proposed a nonparametric spectral density estimator for time-series models with general autocorrelation. Their simulation study showed that the prewhitened kernel estimator reduces bias and mean squared error in spectral density estimation. Christou and Pittis (2002) conducted a Monte Carlo study to investigate the finite-sample properties in the FMOLS procedure. Their results suggest that the prewhitened kernel estimator minimizes the second-order asymptotic bias effects in cointegration regression.

In our Stata `lrcov` and `cointreg` commands, the default specification is the nonparametric kernel method. Users can choose to prewhiten the data or not.

3 lrcov: Stata command to compute LRCOV

`lrcov` computes LRCOV in Mata. For the prewhitened kernel method, matrix inversion is required. `lrcov` uses the `invsym` and `pinv` commands.

3.1 Syntax

`lrcov` *varlist* [*if*] [*in*] [, `wvar(`*varname*`)` `nocenter` `constant` `dof(`#`)` `vic(`*string*`)` `vlag(`#`)` `kernel(`*string*`)` `bwidth(`#`)` `bmeth(`*string*`)` `blag(`#`)` `bweig(`*numlist*`)` `bwmax(`#`)` `btrunc` `disp(`*string*`)`]

varlist may contain factor variables or time-series operators.

3.2 Options

`wvar(`*varname*`)` specifies the weight of the observation; that is, multiply each variable in *varlist* with *varname*.

`nocenter` requests that `lrcov` not center the data before computing. By default, `lrcov` centers the data using the mean before computing LRCOV.

`constant` adds constant to *varlist*. This option may only be used with the `wvar()` option.

`dof(`#`)` adjusts the LRCOV by degrees of freedom. The default is `dof(0)`.

`vic(`*string*`)` specifies the information criteria to select the optimal lags in VAR. `aic`, `bic`, and `hq` are allowed. To prewhiten the data, both `vic()` and `vlag()` must be specified.

`vlag(`#`)` specifies the maximum lag to select the optimal lag length if the `vic()` option is specified. Otherwise, # is the lag order of the VAR model to estimate. If the user specified `vic()` but not `vlag()`, `lrcov` automatically sets the maximum lag to $\text{int}(T^{1/3})$. To prewhiten the data, both `vic()` and `vlag()` must be specified.

`kernel(`*string*`)` specifies the type of kernel function. *string* may be `none`, `bartlett`, `bohman`, `daniell`, `parzen`, `qs`, `priesz`, `pcauchy`, `pgeometric`, `thamming`, `thanning`, or `tparzen`. If the user specifies `kernel(none)`, the `bwidth()`, `bmeth()`, `blag()`, `bweig()`, and `btrunc` options will be ignored.

`bwidth(`#`)` specifies the bandwidth by hand. If this option is specified, the program will ignore the `bmeth()`, `blag()`, `bweig()`, and `btrunc` options.

`bmeth(`*string*`)` specifies the bandwidth selection procedure, including `nwfixed` [Newey–West fixed lag, that is, $4 \times (T/100)^{2/9}$], `andrews`, and `neweywest`. The default is `bmeth(nwfixed)`.

`blag(`#`)` specifies the parameter of n in bandwidth selection (4). If this option is not specified, the program will set it based on $20(n/100)^r$, where (x) means the largest integer less than x and r depends on the kernel function.

`bweig(`*numlist*`)` specifies the weight vector w when automatically computing the bandwidth according to (2) and (3). The number of elements in *numlist* should be equal to the number of variables in *varlist*. The default weight is 1 for all variables.

`bwmax(`#`)` specifies the maximum bandwidth. If the bandwidth supplied by the user or automatically determined by the procedure is greater than #, then `lrcov` will use # as the bandwidth.

`btrunc` truncates the bandwidth to an integer.

`disp(`*string*`)` requests that `lrcov` display the detailed results, including `two` (two-sided LRCOV), `one` (one-sided LRCOV), `sone` (strict one-sided LRCOV), and `cont` (contemporaneous covariance). The default is `disp(two)`.

3.3 Saved results

`lrcov` saves the following in `r()`:

Scalars			
`r(bwidth)`	bandwidth	`b(vlag)`	lag of VAR model
Macros			
`r(kernel)`	kernel function	`r(vic)`	type of information criterion
`r(bmeth)`	automatic bandwidth method		
Matrices			
`r(Omega)`	two-sided LRCOV	`r(Omega0)`	contemporaneous covariance
`r(Omegaone)`	one-sided LRCOV (lag)	`r(Omegasone)`	strict one-sided LRCOV (lag)

3.4 Examples

We use the macroeconomic data downloaded from Stata's official website to illustrate the use of `lrcov`. The data include macroeconomic indicators of industrial production index, `ipman`; an aggregate weekly hours index, `hours`; aggregate unemployment, `unemp`; and real disposable income, `income`.

```
. webuse dfex
(St. Louis Fed (FRED) macro data)

. describe

Contains data from http://www.stata-press.com/data/r12/dfex.dta
  obs:           443                          St. Louis Fed (FRED) macro data
 vars:             6                          14 May 2011 17:59
 size:        19,492
--------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
--------------------------------------------------------------------------------
month           float  %tm                    Month
unemp           double %10.0g                 Civilian unemployment rate
hours           double %10.0g                 Aggregate weekly hours worked
                                                index: total private industries
inc96           double %10.0g                 Real disposable personal income
ipman           double %10.0g                 Industrial production;
                                                manufacturing (NAICS)
income          double %10.0g                 Real disposable income (100´s)
--------------------------------------------------------------------------------
Sorted by:  month
```

Examples of different specifications

We assume the variables are $I(1)$, so we compute the LRCOV of the differenced series directly.

Default case: Bartlett kernel. No prewhitening, Newey–West automatic bandwidth selection.

```
. lrcov d.(ipman income hours unemp)

Long Run Covariance:

VAR Pre-whitening      =  no
Kernel type            =  Bartlett
Bandwidth (Newey-West) =  16.896
Dof adjustment         =  0
```

Two-sided	D. ipman	D. income	D. hours	D. unemp
D.ipman	.9930469	.1527449	.5334321	-.2557916
D.income	.1527449	.0889101	.0795833	-.0443571
D.hours	.5334321	.0795833	.3991929	-.1804581
D.unemp	-.2557916	-.0443571	-.1804581	.0967915

The header consists of `VAR Pre-whitening` (or it could have `VAR Lag` and the information criterion: AIC, Bayesian, or Hannan and Quinn), `Kernel type`, `Bandwidth` and its automatic selection method (Andrews, Newey–West, or N–W fixed), and `Dof adjustment`. The row title of the matrix depends on the specification of the `disp()` option (`disp(two)` by default).

Case 2: Nonparametric kernel approach. Quadratic spectral kernel, fixed bandwidth at 10.

```
. lrcov d.(ipman income hours unemp), kernel(qs) bwidth(10)
Long Run Covariance:
VAR Pre-whitening      =  no
Kernel type            =  Quadratic Spectral
Bandwidth (user)       =  10
Dof adjustment         =  0
```

Two-sided	D. ipman	D. income	D. hours	D. unemp
D.ipman	.9734461	.1399095	.5141061	-.2525444
D.income	.1399095	.0768796	.0741796	-.0403469
D.hours	.5141061	.0741796	.3711272	-.176326
D.unemp	-.2525444	-.0403469	-.176326	.0963466

Case 3: Parametric approach. VARHAC estimation using VAR(1).

```
. lrcov d.(ipman income hours unemp), vlag(1) kernel(none)
Long Run Covariance:
Var lag (user)         =  1
Kernel type            =  None
Dof adjustment         =  0
```

Two-sided	D. ipman	D. income	D. hours	D. unemp
D.ipman	.5507148	.0803142	.2716553	-.131211
D.income	.0803142	.165078	.0401646	-.018165
D.hours	.2716553	.0401646	.1962499	-.0792629
D.unemp	-.131211	-.018165	-.0792629	.0503728

Case 4: Prewhitened kernel approach. Prewhiten using VAR(1), Parzen kernel, and Andrews automatic bandwidth.

```
. lrcov d.(ipman income hours unemp), vlag(1) kernel(parzen) bmeth(andrews)
Long Run Covariance:
Var lag (user)         =  1
Kernel type            =  Parzen
Bandwidth (Andrews)    =  3.8986
Dof adjustment         =  0
```

Two-sided	D. ipman	D. income	D. hours	D. unemp
D.ipman	.5298245	.0898282	.2534481	-.1231246
D.income	.0898282	.1415811	.0394046	-.0187237
D.hours	.2534481	.0394046	.1764541	-.0760452
D.unemp	-.1231246	-.0187237	-.0760452	.0473085

Case 5: More flexible options. Prewhiten using VAR with lag selected by AIC, quadratic spectral kernel, Newey–West automatic bandwidth with truncated lag = 10.

```
. lrcov d.(ipman income hours unemp), vic(aic) kernel(qs) bmeth(neweywest)
> blag(10)
Long Run Covariance:
Var lag (AIC)          =  7
Kernel type            =  Quadratic Spectral
Bandwidth (Newey-West) =  11.958
Dof adjustment         =  0
```

Two-sided	D. ipman	D. income	D. hours	D. unemp
D.ipman	1.47109	.2149076	.8322045	-.4151169
D.income	.2149076	.0898623	.1285044	-.0700764
D.hours	.8322045	.1285044	.6190196	-.2999706
D.unemp	-.4151169	-.0700764	-.2999706	.1593269

Using lrcov to compute HAC variance

Next we use `lrcov` to compute the robust covariance matrix in linear regression, $\mathbf{y} = \mathbf{X}\boldsymbol{\beta}+\mathbf{u}, \mathrm{Var}(\mathbf{u}) = \boldsymbol{\Omega}$. The ordinary least squares (OLS) estimators are $\widehat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$, and their covariance is $\mathrm{Cov}\left(\widehat{\boldsymbol{\beta}}\right) = (\mathbf{X}'\mathbf{X})^{-1}(\mathbf{X}'\boldsymbol{\Omega}\mathbf{X})(\mathbf{X}'\mathbf{X})^{-1}$. We assume the equation to be arbitrarily specified as

$$\Delta\texttt{ipman}_t = \beta_0 + \beta_1\Delta\texttt{income}_t + \beta_2\Delta\texttt{hours}_t + \beta_3\Delta\texttt{unemp}_t + u_t$$

The HAC covariance matrix is computed as follows:

```
. quietly regress d.ipman d.(income hours unemp)
. quietly predict u, res
. quietly lrcov d.(income hours unemp), wvar(u) constant dof(4) kernel(none)
. matrix covu = r(Omega)
. matrix accum xx = d.(income hours unemp)
(obs=442)
. matrix xxi = invsym(xx)
. matrix cov = 442*xxi*covu*xxi
. matlist cov
```

	D. income	D. hours	D. unemp	_cons
D.income	.0021644	-.0005001	.0006004	1.16e-06
D.hours	-.0005001	.0051949	.0021574	-.0011289
D.unemp	.0006004	.0021574	.014977	-.000245
_cons	1.16e-06	-.0011289	-.000245	.0006414

In fact, the `hacreg` command described below computes the robust covariance matrix in just the same way. The same results can be obtained using the following Stata commands:

```
. quietly regress d.ipman d.(income hours unemp), vce(robust)
. matlist e(b)
  (output omitted)
. matlist e(V)
  (output omitted)
```

Using lrcov to perform GMM estimation

Similar computation is easily applied to GMM estimation. The GMM estimator and its variance are

$$\widehat{\boldsymbol{\beta}}_{\text{GMM}} = (\mathbf{X}'\mathbf{Z}\mathbf{W}\mathbf{Z}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Z}\mathbf{W}\mathbf{Z}'\mathbf{y}$$
$$\text{Cov}\left(\widehat{\boldsymbol{\beta}}_{\text{GMM}}\right) = n(\mathbf{X}'\mathbf{Z}\mathbf{W}\mathbf{Z}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{W}\widehat{\mathbf{S}}\mathbf{W}\mathbf{Z}'\mathbf{X}(\mathbf{X}'\mathbf{Z}\mathbf{W}\mathbf{Z}'\mathbf{X})^{-1}$$

where $\mathbf{X}$ are explanatory variables and $\mathbf{Z}$ are instrumental variables. $\widehat{\mathbf{S}}$, the estimator of $E(\mathbf{z}_i u_i u_i \mathbf{z}_i')$, is calculated using the residuals based on $\widehat{\boldsymbol{\beta}}_{\text{GMM}}$. The weight matrix $\mathbf{W} = (\mathbf{Z}'\boldsymbol{\Omega}\mathbf{Z})^{-1}$ is calculated using the residuals from the initial two-stage least-squares estimates. If we set $\mathbf{W} = \widehat{\mathbf{S}}^{-1}$, then we obtain the optimal two-step GMM estimator, and the covariance matrix reduces to $\text{Cov}\left(\widehat{\boldsymbol{\beta}}_{\text{GMM}}\right) = n(\mathbf{X}'\mathbf{Z}\mathbf{W}\mathbf{Z}'\mathbf{X})^{-1}$. The following commands estimate the two-step GMM estimator using the `lrcov` command:

```
. * one-step GMM (two-stage least squares)
. qui ivregress 2sls d.ipman d.income (d.hours d.unemp = DL(1/2).(hours unemp))
. qui predict u, residuals
. * weighted matrix using long run variance
. local inst = "d.income DL(1/2).(hours unemp)"
. qui lrcov `inst´, nocenter wvar(u) constant kernel(bartlett) bwidth(11)
. matrix w = r(Omega)
. * two-step GMM estimator
. qui matrix accum xz = d.hours d.unemp d.income `inst´
. matrix xz = xz[1..3, 4...] \ xz["_cons", 4...]
. matrix accum yz = d.ipman `inst´
(obs=440)
. matrix yz = yz[1, 2...]
. matrix b = invsym(xz*invsym(w)*xz´)*(xz*invsym(w)*yz´)
. matrix V = 440*invsym(xz*invsym(w)*xz´)
. matlist b´
```

	D. hours	D. unemp	D. income	_cons
D.ipman	.5323454	-1.933122	-.0445388	.1229558

```
. matlist V

             |        D.         D.         D.
             |     hours      unemp     income      _cons
-------------+--------------------------------------------
     D.hours |  .0189107
     D.unemp |  .0066483   .1128266
    D.income |  .0010752  -.0003729   .0027054
       _cons | -.0025717  -.0001957  -.0003734   .0009218
```

The same results can be obtained using the `ivregress` command.[1]

```
. qui ivregress gmm d.ipman d.income (d.hours d.unemp = DL(1/2).(hours unemp)),
> vce(unadjusted) wmatrix(hac bartlett 10)
. matlist e(b)
  (output omitted)
. matlist e(V)
  (output omitted)
```

Note that in the Bartlett (Parzen, Parzen–Riesz, etc.) kernels, $k(1) = 0$ and $\text{ceil}(b_T) - 1$ autocovariances enter the estimator with nonzero weights where $\text{ceil}(x)$ denotes the largest integer that is smaller than or equal to x. So the truncation parameter 10 in `ivregress` is equivalent to `bwidth(11)` in `lrcov`.

4 hacreg: HAC standard errors in linear regression

We provide the `hacreg` command to implement the HAC-type standard errors. `hacreg` has several improvements over the official Stata `newey` command. `hacreg` can automatically determine the optimal lag based on information criteria. Moreover, `hacreg` allows more-flexible treatment with LRCOV, such as prewhitening the data and more kernel functions.

4.1 Syntax

`hacreg` *depvar indepvars* [*if*] [*in*] [, <u>noc</u>`onstant` <u>l</u>`evel(#)` *lrcov_options*]

depvar may contain time-series operators. *indepvars* may contain factor variables and time-series operators. `by` is allowed.

1. As pointed out by StataCorp (2009), many software packages that implement GMM estimation use the heteroskedasticity-consistent weighting matrix to obtain the optimal two-step estimates but do not use a heteroskedasticity-consistent variance, even though they may label the standard errors as being robust. To replicate results obtained from other packages, you may have to use the `vce(unadjusted)` option.

4.2 Options

`noconstant` suppresses the constant in the regression equation.

`level(#)` sets the confidence level; default is `level(95)`.

lrcov_options specifies the options to compute LRCOV, which include `kernel(`*string*`)`, `vlag(#)`, `vic(`*string*`)`, `bwidth(#)`, `bmeth(`*string*`)`, `blag(#)`, and `btrunc`. All of these options are specified in the same way as for the `lrcov` command described in section 3.2.

4.3 Saved results

`hacreg` saves the following in `e()`:

Scalars			
`e(N)`	number of observations	`e(df_m)`	model degrees of freedom
`e(r2)`	R-squared	`e(ll)`	log likelihood
`e(r2_a)`	adjusted R-squared	`e(ll_0)`	log likelihood, constant-only model
`e(rank)`	rank of `e(V)`		
`e(rss)`	residual sum of squares	`e(mss)`	residual sum of squares
`e(rmse)`	root of mean squared error	`e(F)`	model F statistic
`e(df_r)`	residual degrees of freedom		
Macros			
`e(cmd)`	`hacreg`	`e(vcetype)`	type of covariance
`e(cmdline)`	command as typed	`e(title)`	title of regression
`e(depvar)`	name of dependent variable	`e(properties)`	`b V`
`e(predict)`	program to implement `predict`		
Matrices			
`e(b)`	coefficient vector	`e(V)`	variance–covariance matrix of the estimators
Functions			
`e(sample)`	marks estimation sample		

4.4 Example: Dynamic impact of cold weather on orange juice price

Stock and Watson (2006) discussed the effect of cold weather on the price change of orange juice using the HAC standard error. The model was specified as

$$\mathtt{dlnpoj}_t = \beta_0 + \sum_{l=1}^{18} \beta_l \mathtt{FDD}_{t-l} + u_t$$

where `dlnpoj` is the price change computed as differencing the log series. `FDD` is the number of freezing degree days. The β_l reflects the dynamic multiplier at lag l. The accumulated dynamic multiplier at lag l can be estimated by the model

$$\mathtt{dlnpoj}_t = \beta_0 + \sum_{l=1}^{17} \beta_l \Delta\mathtt{FDD}_{t-l} + \beta_{18}\mathtt{FDD}_{t-18} + u_t$$

The data are downloadable from Stock and Watson's (2006) website.[2] To replicate their results listed in table 13-1, we choose the same truncation lag in the Bartlett kernel. Note that the truncation parameter m in Stock and Watson (2006) is equivalent to $m+1$ in `hacreg`.

```
. use stockwatson, clear
. generate lnpoj=ln(poj)
. generate dlnpoj=D.lnpoj*100
(1 missing value generated)
. hacreg dlnpoj L(0/18).fdd if tin(1950m1,2000m12), kernel(bartlett) bwidth(8)
      Source |       SS       df       MS         Number of obs     =        612
-------------+------------------------------      F(19, 592)        =   2.257877
       Model |  2014.484439    19    106.0255     Prob > F          =     0.0018
    Residual |  13662.10622   592     23.0779     R-square          =   .1285027
-------------+------------------------------      Adjusted R2       =   .1005324
       Total |  15676.59066   611 25.65726785     Standard error    =   4.803944

------------------------------------------------------------------------------
             |                HAC
      dlnpoj |      Coef.   Std. Err.       t     P>|t|      [95% Conf. Interval]
-------------+----------------------------------------------------------------
         fdd |
         --. |   .5037985   .1395634     3.61    0.000      .2296989    .7778981
         L1. |   .1699179   .0889433     1.91    0.057     -.004765     .3446008
         L2. |   .0670143   .0606926     1.10    0.270     -.0521847    .1862133
         L3. |   .0710866   .0448936     1.58    0.114     -.0170836    .1592567
         L4. |   .0247764    .031656     0.78    0.434     -.0373953    .0869482
         L5. |   .0319348   .0307631     1.04    0.300     -.0284833    .0923528
         L6. |   .0325602   .0476017     0.68    0.494     -.0609285    .1260489
         L7. |   .0149134   .0157426     0.95    0.344     -.0160048    .0458316
         L8. |  -.0421964   .0348847    -1.21    0.227     -.1107094    .0263165
         L9. |  -.0102996   .0514516    -0.20    0.841     -.1113495    .0907503
        L10. |  -.1163004   .0706558    -1.65    0.100     -.2550669    .0224662
        L11. |  -.0662832   .0530143    -1.25    0.212     -.1704023    .0378359
        L12. |  -.1422677   .0774238    -1.84    0.067     -.2943265    .0097911
        L13. |  -.0815754   .0429925    -1.90    0.058     -.1660117     .002861
        L14. |  -.0563725   .0352999    -1.60    0.111     -.1257008    .0129557
        L15. |  -.0318753   .0280183    -1.14    0.256     -.0869027     .023152
        L16. |  -.0067771   .0557013    -0.12    0.903     -.1161733     .102619
        L17. |   .0013941    .018445     0.08    0.940     -.0348315    .0376197
        L18. |   .0018238   .0169734     0.11    0.914     -.0315117    .0351592
             |
       _cons |  -.3402371   .2736588    -1.24    0.214     -.8776974    .1972231
------------------------------------------------------------------------------
```

The accumulated multiplier for different bandwidth and monthly indicators can be estimated as follows:

```
. qui hacreg dlnpoj DL(0/17).fdd L18.fdd if tin(1950m1,2000m12),
> kernel(bartlett) bwidth(8)
. estimates store est2
. qui hacreg dlnpoj DL(0/17).fdd L18.fdd if tin(1950m1,2000m12),
> kernel(bartlett) bwidth(15)
. estimates store est3
```

2. http://wps.aw.com/aw_stock_ie_2/50/13016/3332229.cw/index.html

```
. generate month=month(dofm(mdate))
. qui hacreg dlnpoj DL(0/17).fdd L18.fdd i.month if tin(1950m1,2000m12),
> kernel(bartlett) bwidth(8)
. estimates store est4
```

We view all the results by typing

```
. estimates table est2 est3 est4, b(%6.2f) se(%6.2f) style(oneline)
  (output omitted)
```

Of course, the same results can be obtained with the Stata `newey` command. For example, to fit the last model with monthly dummy variables, type

```
. newey dlnpoj DL(0/17).fdd L18.fdd i.month if tin(1950m1,2000m12), lag(7)
  (output omitted)
```

If you want to obtain the White heteroskedastic robust standard errors, specify `kernel(none)` in the `hacreg` command. In the output table, `hacreg` also reports the variance analysis table based on OLS for reference.

5 cointreg: Cointegration regression based on LRCOV

The study of cointegrating relationships has been a particularly active area of econometric research. Consider the time-series vector process $(y_t, \mathbf{x}_t')'$ with cointegrating relationships

$$\begin{aligned} y_t &= \mathbf{x}_t'\beta + \mathbf{d}_{1t}'\gamma_1 + \mathbf{u}_{1t} \\ \mathbf{x}_t &= \boldsymbol{\Gamma}_1\mathbf{d}_{1t} + \boldsymbol{\Gamma}_2\mathbf{d}_{2t} + \varepsilon_t \\ \Delta\varepsilon_t &= \mathbf{u}_{2t} \end{aligned}$$

where $\mathbf{d}_{1t}$ and $\mathbf{d}_{2t}$ are deterministic trend regressors. $\mathbf{d}_{1t}$ enters into both the cointegration equation and the regressors equations. $\mathbf{d}_{2t}$ only enters into the regressors equations. u_{1t} is the cointegrating equation error. $\mathbf{u}_{2t}$ are regressors innovations.

Assume the innovations $\mathbf{u}_t = (u_{1t}, \mathbf{u}_{2t}')'$ are strictly stationary and ergodic with zero means, contemporaneous covariance matrix $\boldsymbol{\Sigma}$, one-sided LRCOV matrix $\boldsymbol{\Lambda}$, and nonsingular LRCOV matrix $\boldsymbol{\Omega}$.

$$\begin{aligned} \boldsymbol{\Sigma} = E(\mathbf{u}_t{\mathbf{u}_t}') &= \begin{bmatrix} \sigma_{11} & \boldsymbol{\sigma}_{12} \\ \boldsymbol{\sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{bmatrix} \\ \boldsymbol{\Lambda} = \sum\nolimits_{j=0}^{\infty} E(\mathbf{u}_t{\mathbf{u}_{t-j}}') &= \begin{bmatrix} \lambda_{11} & \boldsymbol{\lambda}_{12} \\ \boldsymbol{\lambda}_{21} & \boldsymbol{\Lambda}_{22} \end{bmatrix} \\ \boldsymbol{\Omega} = \sum\nolimits_{j=-\infty}^{\infty} E(\mathbf{u}_t{\mathbf{u}_{t-j}}') &= \begin{bmatrix} \omega_{11} & \boldsymbol{\omega}_{12} \\ \boldsymbol{\omega}_{21} & \boldsymbol{\Omega}_{22} \end{bmatrix} \end{aligned} \tag{5}$$

If the series are cointegrated, then the OLS estimator is consistent, converging at a faster rate than standard. But when there exists long-run correlation between u_{1t} and

$\mathbf{u}_{2t}$ ($\boldsymbol{\omega}_{12}$), or cross-correlation between the cointegration equation error and the regressor innovations ($\boldsymbol{\lambda}_{12}$), then the OLS estimators have an asymptotic distribution that is generally non-Gaussian, asymptotically biased, asymmetric, and involves nonscalar nuisance parameters. So the conventional testing procedures are not valid. Three fully efficient estimation methods—FMOLS (Phillips and Hansen 1990), CCR (Park 1992), and DOLS (Saikkonen 1992; Stock and Watson 1993)—are proposed to get fully efficient estimation.

5.1 FMOLS, DOLS, and CCR cointegration regression

Phillips and Hansen (1990) proposed the FMOLS estimator and Park (1992) proposed the CCR estimator, both of which use a semiparametric correction to eliminate the problems stated above. The estimators are asymptotically unbiased and have fully efficient normal asymptotics, allowing for standard Wald tests using asymptotic chi-squared statistical inference. Let $\widehat{\boldsymbol{\omega}}_{12}$, $\widehat{\boldsymbol{\Omega}}_{22}$, $\widehat{\boldsymbol{\lambda}}_{12}$, and $\widehat{\boldsymbol{\Lambda}}_{22}$ be the corresponding parts of the LRCOV of $\widehat{\mathbf{u}}_t = (\widehat{u}_{1t}, \widehat{\mathbf{u}}_{2t}')'$ according to (5). The FMOLS and CCR estimators can be obtained by transforming the regressors and regressand and then applying the OLS procedure. FMOLS estimation only transforms the regressand

$$y_t^+ = y_t - \widehat{\boldsymbol{\omega}}_{12}\widehat{\boldsymbol{\Omega}}_{22}^{-1}\widehat{\mathbf{u}}_{2t}$$

where $\widehat{u}_{1t}$ is the residual of the cointegration equation estimated by OLS, and $\widehat{\mathbf{u}}_{2t}$ are the differenced residuals of regressor equations or the residuals of the differenced regressor equations.

The FMOLS estimators and their covariance are given by

$$\widehat{\boldsymbol{\theta}} = \begin{bmatrix} \widehat{\boldsymbol{\beta}} \\ \widehat{\gamma}_{\mathbf{1}} \end{bmatrix} = \left[\sum_{t=1}^{T} \mathbf{z}_t\mathbf{z}_t'\right] \left[\sum_{t=1}^{T} \mathbf{z}_t y_t^+ - T\begin{pmatrix} \widehat{\boldsymbol{\lambda}}_{12}^{+'} \\ 0 \end{pmatrix}\right]$$

$$\mathrm{Var}\left(\widehat{\boldsymbol{\theta}}\right) = \widehat{\boldsymbol{\omega}}_{1,2}\left[\sum_{t=1}^{T} \mathbf{z}_t\mathbf{z}_t'\right], \quad \widehat{\boldsymbol{\omega}}_{1,2} = \widehat{\omega}_{11} - \widehat{\boldsymbol{\omega}}_{12}\widehat{\boldsymbol{\Omega}}_{22}^{-1}\widehat{\boldsymbol{\omega}}_{21}$$

where $\widehat{\boldsymbol{\lambda}}_{12}^{+} = \widehat{\boldsymbol{\lambda}}_{12} - \widehat{\boldsymbol{\omega}}_{12}\widehat{\boldsymbol{\Omega}}_{22}^{-1}\widehat{\boldsymbol{\Lambda}}_{22}$ are called bias-correction terms. $\mathbf{z}_t = (x_t', d_{1t}')'$. $\widehat{\omega}_{1,2}$ is the estimate of the LRCOV of u_{1t} conditional on $\mathbf{u}_{2t}$.

The CCR estimation transforms both the regressand and the regressors

$$y_t^+ = y_t - \left\{\widehat{\boldsymbol{\Sigma}}^{-1}\widehat{\boldsymbol{\Lambda}}_2\widetilde{\boldsymbol{\beta}} + \begin{pmatrix} 0 \\ \widehat{\boldsymbol{\Omega}}_{22}^{-1}\widehat{\boldsymbol{\omega}}_{21} \end{pmatrix}\right\}'\widehat{\mathbf{u}}_t$$

$$\mathbf{x}_t^+ = \mathbf{x}_t - \left(\widehat{\boldsymbol{\Sigma}}^{-1}\widehat{\boldsymbol{\Lambda}}_2\right)'\widehat{\mathbf{u}}_t$$

where $\widehat{\boldsymbol{\Lambda}}_2 = (\widehat{\boldsymbol{\Lambda}}_{12}, \widehat{\boldsymbol{\Lambda}}_{22}')'$. $\widetilde{\boldsymbol{\beta}}$ is some consistent estimator of $\boldsymbol{\beta}$, such as the OLS estimator.

The DOLS estimators are obtained by adding the lead and lag of $\Delta \mathbf{x}_t$ to soak up the long-run correlation between u_{1t} and $\mathbf{u}_{2t}$.

$$y_t = \mathbf{x}_t'\boldsymbol{\beta} + \mathbf{d}_{1t}'\boldsymbol{\gamma}_1 + \sum_{j=-q}^{r} \Delta \mathbf{x}_{t+j}'\boldsymbol{\delta} + v_{1t} \tag{6}$$

The OLS estimators of the above equation have the same asymptotic distribution as do FMOLS and CCR. The covariance of these estimators can be computed with the HAC method or by rescaling the ordinary covariance matrix with the regression variance replaced by the LRCOV of $\widehat{v}_{1t}$.

The FMOLS and CCR estimators need both the two-sided and the one-sided LRCOV of $\widehat{\mathbf{u}}_t$, and the DOLS estimators need only the two-sided LRCOV of $\widehat{v}_{1t}$.

5.2 Syntax

`cointreg` *depvar indepvars* [*if*] [*in*] [, `est(`*method*`)` `noconstant` `eqtrend(`*#*`)` `eqdet(`*varlist*`)` `xtrend(`*#*`)` `xdet(`*varlist*`)` `diff` `stage(`*#*`)` `nodivn` `dlead(`*#*`)` `dlag(`*#*`)` `dic(`*string*`)` `dmaxorder(`*#*`)` `dvar(`*varlist*`)` `dvce(`*string*`)` `level(`*#*`)` *lrcov_options*]

depvar may contain time-series operators. *indepvars* may contain time-series operator and factor variables.

5.3 Options

`est(`*method*`)` specifies the estimation method, which can be `fmols`, `dols`, or `ccr`. The default is `est(fmols)`.

`noconstant` suppresses the constant in the cointegration equation. If this option is specified, `eqtrend()` will set to -1 automatically; that is, there is no deterministic term in the cointegration equation.

`eqtrend(`*#*`)` specifies the trend order in the cointegration equation. `eqtrend(0)` denotes the constant term, `eqtrend(1)` denotes the linear trend, and `eqtrend(2)` denotes the quadratic trend. The default is `eqtrend(0)`. A negative value means that there are no deterministic terms. The specification implies all trends up to the specified order, so `eqtrend(2)` means the trend terms include a constant and a linear trend term along with the quadratic term.

`eqdet(`*varlist*`)` specifies the additional deterministic terms in the cointegration equation.

`xtrend(`*#*`)` specifies the trend order in the independent variables. This option is used only for FMOLS and CCR regression. `xtrend(0)`, `xtrend(1)`, and `xtrend(2)` are allowed and have the same meaning as `eqtrend()`. This trend order should be greater than or equal to the order in the `eqtrend()` option; if that requirement is not met, the program will force the two options to be equal.

`xdet(`*varlist*`)` specifies the additional deterministic terms in the independent variables. This option is used for FMOLS and CCR regression.

`diff` obtains $\widehat{u}_{2t}$ by regressing the differenced equation. The default is regressing the equation first and then differencing the residuals.

`stage(`#`)` is used for FMOLS or CCR regression. This option specifies the number to repeat the estimation process, each time using new residuals to compute the LRCOV. The default is `stage(1)`, which performs FMOLS (or CCR) estimation once. For example, `stage(2)` indicates that `cointreg` use the FMOLS (or CCR) residual $\widehat{u}_{1t}$ to recompute LRCOV and estimate the cointegration equation again.

`nodivn` specifies that the program not divide the LRCOV by n in the intermediate steps. Thus this option omits the adjustment of degrees of freedom.

`dlead(`#`)` sets the lead order in DOLS. The default is `dlead(1)`.

`dlag(`#`)` sets the lag order in DOLS. The default is `dlag(1)`. If the number is negative, for example, `dlag(-1)`, `cointreg` will estimate the static ordinary least-squares regression.

`dic(`*string*`)` sets the information criterion used to select optimal lead and lag length in DOLS. *string* can be `aic`, `bic`, or `hq`. If `dic()` is specified, `cointreg` will omit the `dlead()` and `dlag()` options and automatically select the optimal lead (lag).

`dmaxorder(`#`)` sets the maximum length to select optimal lead and lag length in DOLS. The default is set to $\mathrm{int}[\min\{(T-K)/3, 12\} \times (T/100)^{1/4}]$.

`dvar(`*varlist*`)` specifies the variables of X_t whose differenced variables ΔX_t are added in (6). `cointreg` automatically adds the lead and lag terms of all independent variables. This option gives the user the freedom to add his or her own variables in the cointegration equation.

`dvce(`*string*`)` sets the type of covariance matrix in DOLS regression. *string* can be `rescaled`, `hac`, or `ols`. The default is `dvce(rescaled)`.

`level(`#`)` sets the confidence level; default is `level(95)`.

lrcov_options specifies the options to compute LRCOV, which include `kernel(`*string*`)`, `vlag(`#`)`, `vic(`*string*`)`, `bwidth(`#`)`, `bmeth(`*string*`)`, `blag(`#`)`, and `btrunc`. All of these options are specified in the same way as for the `lrcov` command described in section 3.2.

5.4 Saved results

`cointreg` saves the following in `e()`:

Scalars

`e(N)`	number of observations	`e(r2)`	R-squared
`e(r2_a)`	adjusted R-squared	`e(rmse)`	standard error
`e(lrse)`	long-run standard error	`e(rss)`	residual sum of squares
`e(tss)`	total sum of squares	`e(eqtrend)`	trend term in equation
`e(xtrend)`	trend term in regressor	`e(bwidth)`	band width
`e(vlag)`	lag in VAR prewhitening	`e(dlead)`	lead length in DOLS
`e(dlag)`	lag length in DOLS		

Macros

`e(cmd)`	`cointreg`	`e(est)`	estimation method
`e(cmdline)`	command as typed	`e(vic)`	information criterion in VAR
`e(kernel)`	kernel type	`e(bmeth)`	bandwidth selection method
`e(depvar)`	name of dependent variable	`e(properties)`	`b V`
`e(dic)`	lag type in DOLS	`e(vcetype)`	variance type in DOLS

Matrices

`e(b)`	coefficient vector	`e(V)`	variance–covariance matrix of the estimators

Functions

`e(sample)`	marks estimation sample

5.5 Examples

Next we will illustrate the FMOLS, CCR, and DOLS cointegration estimation methods using several classical examples.

FMOLS example: The consumption function in the U.S.

We use the example in Hansen (1992) to illustrate FMOLS estimation. The data contain seasonally adjusted aggregate quarterly U.S. consumption (`tc`) and total disposable income (`di`) in real per capita units, for the period 1953:2–1984:4. The data can be downloaded from Hansen's homepage.[3]

A constant and a time trend are included in the equation:

$$\texttt{tc}_t = \beta_0 + \beta_1 t + \beta_2 \texttt{di}_t + u_t$$

We adopt the quadratic spectral kernel and the Andrews automatic bandwidth selection method. The following commands estimate the above equation. To replicate the results of Hansen (1992), the `nodivn` option must be specified.

3. http://www.ssc.wisc.edu/~bhansen/

```
. use campbell, clear
. cointreg tc di, est(fmols) vlag(1) kernel(qs) bmeth(andrews) eqtrend(1) nodivn
Cointegration regression (FMOLS):
VAR lag(user)         =    1                  Number of obs     =        126
Kernel                =    qs                 R2                =   .9947833
Bandwidth(andrews)    =    0.8877             Adjusted R2       =   .9946985
                                              S.e.              =   51.41235
                                              Long run S.e.     =   108.0174
```

tc	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
di	.9818777	.0882267	11.13	0.000	.8089565	1.154799
linear	-1.022808	1.829954	-0.56	0.576	-4.609453	2.563836
_cons	-112.1549	194.2193	-0.58	0.564	-492.8177	268.508

The `linear` in the output table denotes the linear trend in the regression equation.

CCR example: The consumption of nondurable goods

We use the example of Ogaki (1993). The model is specified as

$$\texttt{ndur}_t = \beta_0 + \beta_1 \texttt{price}_t + \beta_2 \texttt{dur}_t + u_t$$

where $\texttt{ndur}_t$ is real consumption of nondurable goods per capital, $\texttt{dur}_t$ is real consumption of durable goods per capital, and $\texttt{price}_t$ is the relative price index of nondurable and durable goods. The data can be downloaded from Ogaki's homepage.[4]

We use the quadratic spectral kernel and the Andrews automatic bandwidth selection method. The three-stage CCR regression is estimated by the following commands:

```
. use ccr
. cointreg ndur price dur, est(ccr) vlag(1) kernel(qs) bmeth(andrews) stage(3)
Cointegration regression (CCR):
VAR lag(user)         =    1                  Number of obs     =        168
Kernel                =    qs                 R2                =   .4551902
Bandwidth(andrews)    =    0.5839             Adjusted R2       =   .4485864
                                              S.e.              =   .1514857
                                              Long run S.e.     =   .0866897
```

ndur	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
price	.5682032	.1385131	4.10	0.000	.2967225	.839684
dur	.5092836	.0448642	11.35	0.000	.4213515	.5972158
_cons	4.590851	.326225	14.07	0.000	3.951462	5.23024

4. http://www.econ.ohio-state.edu/ogaki/

DOLS example: The demand for money in the United States

The money demand equation typically estimated in the literature is specified as

$$\mathtt{m}_t - \mathtt{p}_t = \mu + \theta_y \mathtt{y}_t + \theta_r \mathtt{r}_t + u_t$$

where $\mathtt{m}_t$ is the log of money stock in period t, $\mathtt{p}_t$ is the log of price level, $\mathtt{y}_t$ is log income, $\mathtt{r}_t$ is nominal interest rate, and u_t is the error term. θ_y is income elasticity and θ_r is the interest semielasticity of money demand. Hayashi (2000) cited this example in his textbook; the data are downloadable from his homepage.[5]

All the variables are $I(1)$, and we skip the unit-root testing and directly fit the model using the DOLS method. We fit the model using the full sample and two subsamples with two leads and two lags, in accordance with Stock and Watson (1993). The LRCOV is computed using prewhitening with the AR(2) model. The following commands repeat the results of table 10.2 in Hayashi (2000), which consist of static ordinary least-squares and DOLS estimations.

```
. use sw93
(Source: Stock and Watson(1993))
. qui cointreg mp y r if tin(1903, 1987), est(dols) dlag(-1) kernel(none)
. qui estimates store SOLS
. qui cointreg mp y r, est(dols) dlead(2) dlag(2) vlag(2) kernel(none)
. qui estimates store DOLS
. estimates table SOLS DOLS, b(%6.3f) se(%6.3f) drop(_cons) style(oneline)
```

Variable	SOLS	DOLS
y	0.943	0.970
	0.022	0.046
r	-0.082	-0.101
	0.006	0.013

legend: b/se

For the Chow break-point test at year 1946, Hayashi (2000) fit the model

$$\mathtt{m}_t - \mathtt{p}_t = \mu + \gamma_y \mathtt{y}_t + \gamma_r \mathtt{r}_t + \delta_0 \mathtt{D}_t + \delta_y \mathtt{y}_t \mathtt{D}_t + \delta_r \mathtt{r}_t \mathtt{D}_t + u_t$$

where $\mathtt{D}_t$ is a dummy variable whose value is 1 if $t \geq 1946$ and 0 otherwise. We use factor variables in Stata to fit the model, and we use the standard `test` command to do the Chow test:

5. http://fhayashi.fc2web.com/datasets.htm

```
. generate dum=year>=1946
. qui cointreg mp y r i.dum i.dum#c.(y r), est(dols) dlead(2) dlag(2) vlag(2)
> kernel(none)
. test 1.dum 1.dum#c.y 1.dum#c.r
 ( 1)  1.dum = 0
 ( 2)  1.dum#c.y = 0
 ( 3)  1.dum#c.r = 0
           chi2(  3) =   19.12
         Prob > chi2 =    0.0003
```

Note that only the differenced variables of $(\mathtt{y}_t, \mathtt{r}_t)$ enter into the DOLS equation, so we should have specified the `dvar(y r)` option. Hayashi (2000) listed the results in table 10.4.

6 Conclusion and extension

In time-series econometrics, estimating the long-run variance matrix of a random vector process is essential for empirical research on estimation (for example, GMM and cointegration regression) and testing (for example, HAC standard error, unit root, and cointegration testing) problems. We propose the `lrcov` command to compute LRCOV; `lrcov` includes many kernel functions and allows the user to prewhiten the data. Based on `lrcov`, we provide two other commands, **hacreg** and **cointreg**. **hacreg** estimates HAC standard errors. Compared with the official Stata **newey** command, **hacreg** is more flexible in that it contains more kernel functions, automatically determines the lag order, and prewhitens the data. **cointreg** estimates three cointegration regressions: FMOLS, DOLS, and CCR, all of which need to compute the LRCOV. We use several examples to illustrate these commands.

Many extensions can be made based on our work. More kernels and more bandwidth selection methods can be allowed in LRCOV estimation. Some examples may include further extensions on the `lrcov` command. Phillips, Sun, and Jin (2007) pursued the approach of Kiefer and Vogelsang (2002a,b) and proposed a class of steep origin kernels, which are constructed by exponentiating a mother kernel and can be used without truncation. The steep origin kernels are asymptotically mean squared error equivalent, so choice of mother kernel does not matter asymptotically. Jin, Phillips, and Sun (2006) used steep origin kernels in cointegrated systems, and simulations indicated that robust tests have improved size and power properties.

Hirukawa (2010) suggested a two-stage plug-in bandwidth selection approach that estimates an unknown quantity in the optimal bandwidth for the HAC estimator (called normalized curvature) using a general class of kernels, and derives the optimal bandwidth that minimizes the asymptotic mean squared error of the estimator of normalized curvature. It is shown that the optimal bandwidth for the kernel-smoothed normalized curvature estimator should diverge at a slower rate than that of the HAC estimator using the same kernel. Hirukawa (2011) revealed that the new bandwidth choice rule contributes bias reduction in the estimators for cointegration regression models.

LRCOV for more stochastic processes may be another important aspect for extension. Phillips and Kim (2007) derived an asymptotic expansion for the autocovariance matrix of a vector of stationary long-memory processes and applied the theory to deliver formulas for the LRCOV matrices of multivariate time series with long memory. Abadir, Distaso, and Giraitis (2009) extended the usual Bartlett-kernel HAC estimator to deal with long memory and antipersistence, and derived asymptotic expansions for this estimator and the memory and autocorrelation consistent estimator.

7 Acknowledgments

We would like to thank H. Joseph Newton (the editor) for providing advice and encouragement. Constructive comments and insightful suggestions from an anonymous referee, Jian Yang, and Liuling Li substantially helped the revision of this article. We also thank B. E. Hansen, M. Ogaki, and F. Hayashi. The data in our examples are directly downloaded from their respective websites. Qunyong Wang acknowledges Project "Signal Extraction Theory and Applications of Robust Seasonal Adjustment" (Grant No. 71101075) supported by NSFC and the Center for Experimental Education of Economics at Nankai University.

8 References

Abadir, K. M., W. Distaso, and L. Giraitis. 2009. Two estimators of the long-run variance: Beyond short memory. *Journal of Econometrics* 150: 56–70.

Andrews, D. W. K. 1991. Heteroskedasticity and autocorrelation consistent covariance matrix estimation. *Econometrica* 59: 817–858.

Andrews, D. W. K., and J. C. Monahan. 1992. An improved heteroskedasticity and autocorrelation consistent covariance matrix estimator. *Econometrica* 60: 953–966.

Brillinger, D. R. 1980. *Time Series: Data Analysis and Theory*. New York: Holt, Rinehart and Winston.

Choi, I., J. Y. Park, and B. Yu. 1997. Canonical cointegrating regression and testing for cointegration in the presence of I(1) and I(2) variables. *Econometric Theory* 13: 850–876.

Christou, C., and N. Pittis. 2002. A nonparametric prewhitened covariance estimator. *Econometric Theory* 18: 948–961.

den Haan, W. J., and A. Levin. 1996. Inferences from parametric and non-parametric covariance matrix estimation procedures. NBER Technical Working Paper No. 195. http://www.nber.org/papers/t0195.html.

———. 1997. A practitioner's guide to robust covariance matrix estimation. In *Handbook of Statistics 15: Robust Inference*, ed. G. S. Maddala and C. R. Rao, 291–341. Amsterdam: North-Holland.

———. 2000. Robust covariance matrix estimation with data-dependent VAR prewhitening order. NBER Technical Working Paper No. 255. http://www.nber.org/papers/t0255.

Hall, A. R. 2005. *Generalized Method of Moments*. Oxford: Oxford University Press.

Hansen, B. E. 1992. Tests for parameter instability in regressions with I(1) processes. *Journal of Business and Economic Statistics* 10: 321–335.

Hayashi, F. 2000. *Econometrics*. Princeton, NJ: Princeton University Press.

Hirukawa, M. 2010. A two-stage plug-in bandwidth selection and its implementation for covariance estimation. *Econometric Theory* 26: 710–743.

———. 2011. How useful is yet another data-driven bandwidth in long-run variance estimation? A simulation study on cointegrating regressions. *Economics Letters* 111: 170–172.

Jin, S., P. C. B. Phillips, and Y. Sun. 2006. A new approach to robust inference in cointegration. *Economics Letters* 91: 300–306.

Kiefer, N. M., and T. J. Vogelsang. 2002a. Heteroskedasticity-autocorrelation robust testing using bandwidth equal to sample size. *Econometric Theory* 18: 1350–1366.

———. 2002b. Heteroskedasticity-autocorrelation robust standard errors using the Bartlett kernel without truncation. *Econometrica* 70: 2093–2095.

Lee, C. C., and P. C. B. Phillips. 1994. An ARMA prewhitened long-run variance estimator. Working paper, Yale University. http://korora.econ.yale.edu/phillips/papers/prewhite.pdf.

Marmol, F., and C. Velasco. 2004. Consistent testing of cointegrating relationships. *Econometrica* 72: 1809–1844.

Newey, W. K., and K. D. West. 1987. A simple, positive semi-definite, heteroskedasticity and autocorrelation consistent covariance matrix. *Econometrica* 55: 703–708.

———. 1994. Automatic lag selection in covariance matrix estimation. *Review of Economic Studies* 61: 631–653.

Ogaki, M. 1993. CCR: A user guide. RCER Working Papers 349, University of Rochester, Center for Economic Research. http://ideas.repec.org/p/roc/rocher/349.html.

Park, J. Y. 1992. Canonical cointegrating regressions. *Econometrica* 60: 119–143.

Park, J. Y., and M. Ogaki. 1991. Inference in cointegrated models using VAR prewhitening to estimate shortrun dynamics. RCER Working Papers 281, University of Rochester, Center for Economic Research. http://ideas.repec.org/p/roc/rocher/281.html.

Pedroni, P. 2004. Panel cointegration: Asymptotic and finite sample properties of pooled time series tests with an applicaton to the PPP hypothesis. *Econometric Theory* 20: 597–625.

Phillips, P. C. B., and B. E. Hansen. 1990. Statistical inference in instrumental variables regression with I(1) processes. *Review of Economics Studies* 57: 99–125.

Phillips, P. C. B., and C. S. Kim. 2007. Long-run covariance matrices for fractionally integrated processes. *Econometric Theory* 23: 1233–1247.

Phillips, P. C. B., and P. Perron. 1988. Testing for a unit root in time series regression. *Biometrika* 75: 335–346.

Phillips, P. C. B., Y. Sun, and S. Jin. 2007. Long run variance estimation using steep origin kernels without truncation. *Journal of Statistical Planning and Inference* 137: 985–1023.

Priestley, M. B. 1981. *Spectral Analysis and Time Series*. San Diego: Academic Press.

QMS. 2010. *EViews 7.1 Supplement*. Irvine, CA: Quantitative Micro Software.

Quintos, C. E. 1998. Fully modified vector autoregressive inference in partially nonstationary models. *Journal of the American Statistical Association* 93: 783–795.

Saikkonen, P. 1992. Estimation and testing of cointegrated systems by an autoregressive approximation. *Econometric Theory* 8: 1–27.

StataCorp. 2009. *Stata 11 Base Reference Manual*. College Station, TX: Stata Press.

Stock, J. H., and M. Watson. 1993. A simple estimator of cointegrating vectors in higher order integrated systems. *Econometrica* 61: 783–820.

Stock, J. H., and M. W. Watson. 2006. *Introduction to Econometrics*. 2nd ed. New Jersey: Prentice Hall.

Sul, D., P. C. B. Phillips, and C. Y. Choi. 2005. Prewhitening bias in HAC estimation. *Oxford Bulletin of Economics and Statistics* 67: 517–546.

White, H., and I. Domowitz. 1984. Nonlinear regression with dependent observations. *Econometrica* 52: 143–161.

Xiao, Z., and L. Oliver. 2002. A nonparametric prewhitened covariance estimator. *Journal of Time Series Analysis* 23: 215–250.

About the authors

Qunyong Wang earned his PhD from Nankai University. Now he works at the Institute of Statistics and Econometrics at Nankai University. The article was written as part of his involvement with the Stata research program in China.

Na Wu received her PhD from Nankai University. Now she works at the School of Economics at Tianjin University of Finance and Economics. Her expertise lies in international economics and the financial market.